
pokerthproto Documentation

Release

Florian Wilhelm

May 27, 2015

1	Contents	3
1.1	Writing a Client	3
1.2	Status of Implementation	4
1.3	pokerthproto	7
2	Indices and tables	115
	Python Module Index	117

The goal of this project is to provide a client interface to a [PokerTH](#) server. This interface could then be used to write alternative poker clients in Python or even poker bots.

Right now, this project is in a **beta** status, meaning that it is still incomplete.

1.1 Writing a Client

PokerTHProto is built with [Twisted](#), an event-driven networking engine engine written in Python. That means it is rather useful to have basic knowledge about [event-driven programming](#). But don't get scared, it is pretty easy.

1.1.1 Twisted Application

The easiest way to write a PokerTH client is to write an [Twisted application](#).

This basic `my_client.tac` file gives you an idea:

```
from twisted.application import internet, service
from twisted.internet import reactor

from pokerthproto.protocol import ClientProtocolFactory, ClientProtocol
from pokerthproto.lobby import GameInfo, LobbyError
from pokerthproto.poker import Action

class PyClientProtocol(ClientProtocol):
    def handleInsideLobby(self, lobbyInfo):
        try:
            gameId = self.factory.lobby.getGameInfoId('My Online Game')
        except LobbyError:
            reactor.callLater(1, self.handleInsideLobby, lobbyInfo)
        else:
            self.sendJoinExistingGame(gameId)

    def handleMyTurn(self, gameInfo):
        if gameInfo.highestSet > gameInfo.myBet:
            action, bet = Action.CALL, gameInfo.highestSet
        else:
            action = Action.CHECK
        self.sendMyAction(action, gameInfo.highestSet - gameInfo.myBet)

class PyClientProtocolFactory(ClientProtocolFactory):
    protocol = PyClientProtocol

application = service.Application('PokerTH Client')
```

```
client_factory = PyClientProtocolFactory('PyClient1')
service = internet.TCPClient('localhost', 7234, client_factory)
service.setServiceParent(application)
```

Here, we create an own protocol by inheriting from `ClientProtocol` and overwriting some methods in order to adapt them to our needs. For instance, the `handleInsideLobby` method is triggered when we are inside the lobby. In this case, our action is to join the game named *My Online Game* if available otherwise we wait one second and try again. The method `handleMyTurn` is called during a poker game and here we specified that we want to check if possible and otherwise call. The remaining lines are just boilerplate code to define a Twisted application. You can run this client by calling:

```
twisted -y my_client.tac -n
```

1.1.2 Mandatory Methods

Your own protocol needs to define some mandatory methods:

- *handleInsideLobby*: This method is called when we are inside the lobby. Use *sendJoinExistingGame* or *sendJoinNewGame* to join or create a new game. If you create a new game use *sendStartEvent* to start the game. The `lobbyInfo` argument of type *Lobby* provides you information about other players and games.
- *handleMyTurn*: This method is called when our turn starts. Use *sendMyAction* to decide what action you want to take. The current state of the game is represented with the `gameInfo` parameter of type *Game* in both functions.

1.1.3 Optional Methods

- *handleChat*: This method is called when a chat message was received. Use *sendChatRequest* to reply or start a chat.
- *handleOthersTurn*: This method is called when another player starts its turn. You could use this event to chat him up.
- *handleEndOfHand*: This method is called when one hand is over. Use the `gameInfo` parameter of type *Game* to do logging or learn about the other players' cards and wins with `gameInfo.othersCards` and `gameInfo.wins`.
- *handleEndOfGame*: This method is called when a game is over. Use the `gameInfo` parameter of type *Game* to do logging or learn about the other players' cards and wins with `gameInfo.othersCards` and `gameInfo.wins`. The parameter `winner` of type *Player* provides you the winner of the game. When this function is called you are back in the lobby.

1.2 Status of Implementation

The PokerTH protocol consists of 81 different messages types which are all enveloped inside an `PokerTHMessage`. Only a subset of all messages is needed in most cases.

1.2.1 Implemented

- `AnnounceMessage`
- `InitMessage`

- InitAckMessage
- PlayerListMessage
- GameListNewMessage
- GameListUpdateMessage
- GameListPlayerJoinedMessage
- GameListPlayerLeftMessage
- PlayerInfoRequestMessage
- PlayerInfoReplyMessage
- JoinExistingGameMessage
- JoinNewGameMessage
- JoinGameAckMessage
- GamePlayerJoinedMessage
- GamePlayerLeftMessage
- StartEventMessage
- StartEventAckMessage
- GameStartInitialMessage
- HandStartMessage
- PlayersTurnMessage
- MyActionRequestMessage
- YourActionRejectedMessage
- PlayersActionDoneMessage
- DealFlopCardsMessage
- DealTurnCardMessage
- DealRiverCardMessage
- ChatMessage
- ChatRequestMessage
- AllInShowCardsMessage
- EndOfHandShowCardsMessage
- EndOfHandHideCardsMessage
- ShowMyCardsRequestMessage
- AfterHandShowCardsMessage
- EndOfGameMessage

1.2.2 Not Implemented

- AuthServerChallengeMessage
- AuthClientResponseMessage
- AuthServerVerificationMessage
- AvatarRequestMessage
- AvatarHeaderMessage
- AvatarDataMessage
- AvatarEndMessage
- UnknownAvatarMessage
- GameListAdminChangedMessage
- SubscriptionRequestMessage
- RejoinExistingGameMessage
- JoinGameFailedMessage
- GameAdminChangedMessage
- RemovedFromGameMessage
- KickPlayerRequestMessage
- LeaveGameRequestMessage
- InvitePlayerToGameMessage
- InviteNotifyMessage
- RejectGameInvitationMessage
- RejectInvNotifyMessage
- GameStartRejoinMessage
- PlayerIdChangedMessage (SOON)
- AskKickPlayerMessage
- AskKickDeniedMessage
- StartKickPetitionMessage
- VoteKickRequestMessage
- VoteKickReplyMessage
- KickPetitionUpdateMessage
- EndKickPetitionMessage
- StatisticsMessage
- ChatRejectMessage
- DialogMessage
- TimeoutWarningMessage
- ResetTimeoutMessage
- ReportAvatarMessage

- ReportAvatarAckMessage
- ReportGameMessage
- ReportGameAckMessage
- ErrorMessage
- AdminRemoveGameMessage
- AdminRemoveGameAckMessage
- AdminBanPlayerMessage
- AdminBanPlayerAckMessage
- GameListSpectatorJoinedMessage
- GameListSpectatorLeftMessage
- GameSpectatorJoinedMessage
- GameSpectatorLeftMessage

1.3 pokerthproto

1.3.1 pokerthproto package

Submodules

pokerthproto.game module

All functionality related to a poker game and its representation.

class `pokerthproto.game.ActionInfo` (*player, kind, money=None*)
 Bases: `object`

The action of a player during the poker game.

Parameters

- **player** – player (*Player*)
- **kind** – type of the action (*Action*)
- **money** – stake of the action if available

class `pokerthproto.game.Game` (*gameId, myPlayerId*)
 Bases: `object`

A poker game holding the information about the actions of the players.

addAction (*playerId, kind, money=None*)
 Adds an action to the current round of the game

Parameters

- **playerId** – id of player
- **kind** – type of the action of *Action*
- **money** – stake of the action if available

addOthersCards (*playerId, cards*)

addPlayer (*player*)

addRound (*name*, *cards=None*)

Adds a poker round to the game

Parameters

- **name** – poker round of type *Round*
- **cards** – board cards of the round

addWin (*playerId*, *money*)

bigBlind

currRound

currRoundInfo

Current poker round

Returns poker round

Return type *RoundInfo*

dealer

delPlayer (*player*)

existPlayer (*id*)

Checks if a player exists in the game

Parameters **id** – id of the player

Returns test if player exists

existRound (*name*)

Checks if the poker round exists in this game

Parameters **name** – poker round of *Round*

Returns test if round exists

gameId

getActions (*playerId=None*, *rounds=None*)

Retrieves actions from the game with optional restrictions on rounds and a player.

Parameters

- **playerId** – id of the player or *None* for all players
- **rounds** – list of rounds (*Round*) to consider

Returns list of actions (*ActionInfo*)

getPlayer (*id*)

Retrieves a player from the game

Parameters **id** – id of the player

Returns player

handNum

highestSet

minimumRaise

myBet

othersCards
players
pocketCards
seats
smallBlind
startNewHand()
wins

exception pokerthproto.game.**GameError**

Bases: `exceptions.Exception`

class pokerthproto.game.**RoundInfo** (*gameState*, *cards=None*)

Bases: `object`

Information about the poker round.

Parameters

- **gameState** – name of the poker round (*Round*)
- **cards** – board card of the round as defined in deck

actions

cards

gameState

name

pokerthproto.lobby module

All functionality related to the lobby where information about running games, players etc. is presented.

class pokerthproto.lobby.**GameInfo** (*gameName=None*)

Bases: `object`

Wrapper object for `pokerth_pb2.PNetGameInfo`

This object is needed in order to create an own game.

addPlayer (*player*)

adminPlayerId

allowSpectators

delPlayer (*player*)

delayBetweenHands

endRaiseMode

endRaiseSmallBlindValue

fillWithComputerPlayers

firstSmallBlind

gameId

gameMode

gameName
getMsg ()
isPrivate
manualBlinds
maxNumPlayers
netGameType
playerActionTimeout
players
proposedGuiSpeed
raiseEveryHands
raiseIntervalMode
setInfo (gameInfo)
startMoney

class pokerthproto.lobby.**Lobby**

Bases: `object`

addGameInfo (gameInfo)
addPlayer (playerId)
addPlayerToGame (playerId, gameId)
delPlayer (playerId)
delPlayerFromGame (playerId, gameId)
gameInfos
getGameInfo (gameId)
getGameInfoId (gameName)
getPlayer (playerId)
players
setPlayerInfo (playerId, infoData)

exception pokerthproto.lobby.**LobbyError**

Bases: `exceptions.Exception`

pokerthproto.player module

All functionality related to the representation of a poker player.

class pokerthproto.player.**Player** (*playerId*)

Bases: `object`

Player in poker game including all information of `pokerth_pb2.PlayerInfoReplyMessage.playerInfoData`

avatarHash
avatarType

```

countryCode
isHuman
money
name
playerId
playerRights
seat
setInfo (infoData)

```

pokerthproto.poker module

All data structures related to poker like poker actions, cards, rounds etc.

```
class pokerthproto.poker.Action
```

Bases: `object`

Enum of possible player actions in poker

```
ALLIN = 6
```

```
BET = 4
```

```
CALL = 3
```

```
CHECK = 2
```

```
FOLD = 1
```

```
NONE = 0
```

```
RAISE = 5
```

```
class pokerthproto.poker.Round
```

Bases: `object`

Enum of poker rounds where posting blinds is considered a round too.

```
BIG_BLIND = 5
```

```
FLOP = 1
```

```
PREFLOP = 0
```

```
RIVER = 3
```

```
SMALL_BLIND = 4
```

```
TURN = 2
```

```
pokerthproto.poker.cardToInt (card)
```

Converts a poker card into an integer representation.

Parameters `card` – poker card like 2d, Th, Qc etc.

Returns integer

```
pokerthproto.poker.intToCard (i)
```

Converts an integer into a poker card

Parameters `i` – integer

Returns poker card like 2d, Th, Qc etc.

pokerthproto.pokerth_pb2 module

```
class pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage (**kwargs)
    Bases: google.protobuf.message.Message

    AdminBanPlayerResult = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>
    BANPLAYERID_FIELD_NUMBER = 1
    BANPLAYERRESULT_FIELD_NUMBER = 2
    ByteSize ()
    Clear ()
    ClearField (field_name)
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    FindInitializationErrors ()
        Finds required fields which are not initialized.
        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
            "foo.bar[5].baz".
    static FromString (s)
    HasField (field_name)
    IsInitialized (errors=None)
        Checks if all required fields of a message are set.
        Args:
            errors: A list which, if provided, will be populated with the field paths of all missing required
                fields.
        Returns: True iff the specified message has all required fields set.
    ListFields ()
    MergeFrom (msg)
    MergeFromString (serialized)
    static RegisterExtension (extension_handle)
    SerializePartialToString ()
    SerializeToString ()
    SetInParent ()
        Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
    WhichOneof (oneof_name)
        Returns the name of the currently set field inside a oneof, or None.
    banPlayerAccepted = 0
    banPlayerDBError = 3
    banPlayerId
        Magic attribute generated for "banPlayerId" proto field.
```



```

banPlayerInvalid = 4
banPlayerNoDB = 2
banPlayerPending = 1
banPlayerResult
    Magic attribute generated for “banPlayerResult” proto field.
class pokerthproto.pokerth_pb2.AdminBanPlayerMessage (**kwargs)
    Bases: google.protobuf.message.Message
BANPLAYERID_FIELD_NUMBER = 1
ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.
    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.
static FromString (s)
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.
    Args:
        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.
    Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
banPlayerId
    Magic attribute generated for “banPlayerId” proto field.
class pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage (**kwargs)
    Bases: google.protobuf.message.Message
AdminRemoveGameResult = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

```

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REMOVEGAMEID_FIELD_NUMBER = 1

REMOVEGAMERESULT_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameRemoveAccepted = 0

gameRemoveInvalid = 1

removeGameId

Magic attribute generated for "removeGameId" proto field.

removeGameResult

Magic attribute generated for "removeGameResult" proto field.

class pokerthproto.pokerth_pb2.**AdminRemoveGameMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

```

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
    "foo.bar[5].baz".

static FromString (s)

HasField (field_name)

IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:

        errors: A list which, if provided, will be populated with the field paths of all missing required
        fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (msg)

MergeFromString (serialized)

REMOVEGAMEID_FIELD_NUMBER = 1

static RegisterExtension (extension_handle)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

removeGameId
    Magic attribute generated for "removeGameId" proto field.

class pokerthproto.pokerth_pb2.AfterHandShowCardsMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        "foo.bar[5].baz".

    static FromString (s)

    HasField (field_name)

    IsInitialized (errors=None)
        Checks if all required fields of a message are set.

```

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERRESULT_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

playerResult

Magic attribute generated for “playerResult” proto field.

class pokerthproto.pokerth_pb2.**AllInShowCardsMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

```

PLAYERSALLIN_FIELD_NUMBER = 2
class PlayerAllIn (**kwargs)
    Bases: google.protobuf.message.Message
    ALLINCARD1_FIELD_NUMBER = 2
    ALLINCARD2_FIELD_NUMBER = 3
    ByteSize ()
    Clear ()
    ClearField (field_name)
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    FindInitializationErrors ()
        Finds required fields which are not initialized.
        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message,
            e.g. "foo.bar[5].baz".
    static FromString (s)
    HasField (field_name)
    IsInitialized (errors=None)
        Checks if all required fields of a message are set.
        Args:
            errors: A list which, if provided, will be populated with the field paths of all missing re-
                quired fields.
        Returns: True iff the specified message has all required fields set.
    ListFields ()
    MergeFrom (msg)
    MergeFromString (serialized)
    PLAYERID_FIELD_NUMBER = 1
    static RegisterExtension (extension_handle)
    SerializePartialToString ()
    SerializeToString ()
    SetInParent ()
        Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state
        change.
    WhichOneof (oneof_name)
        Returns the name of the currently set field inside a oneof, or None.
    allInCard1
        Magic attribute generated for "allInCard1" proto field.
    allInCard2
        Magic attribute generated for "allInCard2" proto field.
    playerId
        Magic attribute generated for "playerId" proto field.
    static AllInShowCardsMessage.RegisterExtension (extension_handle)
    AllInShowCardsMessage.SerializePartialToString ()

```

AllInShowCardsMessage.**SerializeToString** ()

AllInShowCardsMessage.**SetInParent** ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

AllInShowCardsMessage.**WhichOneof** (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

AllInShowCardsMessage.**gameId**

Magic attribute generated for “gameId” proto field.

AllInShowCardsMessage.**playersAllIn**

Magic attribute generated for “playersAllIn” proto field.

class pokerthproto.pokerth_pb2.**AnnounceMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

LATESTBETAREVISION_FIELD_NUMBER = 3

LATESTGAMEVERSION_FIELD_NUMBER = 2

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NUMPLAYERSONSERVER_FIELD_NUMBER = 5

PROTOCOLVERSION_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SERVERTYPE_FIELD_NUMBER = 4

SerializePartialToString ()

SerializeToString ()

ServerType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

SetInParent ()
 Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

class Version (**kwargs)
 Bases: `google.protobuf.message.Message`

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
 Finds required fields which are not initialized.
Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)
 Checks if all required fields of a message are set.
Args:
errors: A list which, if provided, will be populated with the field paths of all missing required fields.
Returns: True iff the specified message has all required fields set.

ListFields ()

MAJORVERSION_FIELD_NUMBER = 1

MINORVERSION_FIELD_NUMBER = 2

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
 Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
 Returns the name of the currently set field inside a oneof, or None.

majorVersion
 Magic attribute generated for “majorVersion” proto field.

minorVersion
 Magic attribute generated for “minorVersion” proto field.

`AnnounceMessage`.**WhichOneof** (*oneof_name*)
 Returns the name of the currently set field inside a oneof, or None.

`AnnounceMessage`.**latestBetaRevision**
 Magic attribute generated for “latestBetaRevision” proto field.

`AnnounceMessage.latestGameVersion`

Magic attribute generated for “latestGameVersion” proto field.

`AnnounceMessage.numPlayersOnServer`

Magic attribute generated for “numPlayersOnServer” proto field.

`AnnounceMessage.protocolVersion`

Magic attribute generated for “protocolVersion” proto field.

`AnnounceMessage.serverType`

Magic attribute generated for “serverType” proto field.

`AnnounceMessage.serverTypeInternetAuth = 2`

`AnnounceMessage.serverTypeInternetNoAuth = 1`

`AnnounceMessage.serverTypeLAN = 0`

class `pokerthproto.pokerth_pb2.AskKickDeniedMessage` (**kwargs)

Bases: `google.protobuf.message.Message`

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

KICKDENIEDREASON_FIELD_NUMBER = 3

KickDeniedReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
 Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
 Returns the name of the currently set field inside a oneof, or None.

gameId
 Magic attribute generated for “gameId” proto field.

kickDeniedAlreadyInProgress = 3

kickDeniedInvalidGameState = 0

kickDeniedInvalidPlayerId = 4

kickDeniedNotPossible = 1

kickDeniedReason
 Magic attribute generated for “kickDeniedReason” proto field.

kickDeniedTryAgainLater = 2

playerId
 Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**AskKickPlayerMessage** (**kwargs)
 Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
 Finds required fields which are not initialized.
Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)
 Checks if all required fields of a message are set.
Args:
errors: A list which, if provided, will be populated with the field paths of all missing required fields.
Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

```
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
gameId
    Magic attribute generated for “gameId” proto field.
playerId
    Magic attribute generated for “playerId” proto field.
class pokerthproto.pokerth_pb2.AuthClientResponseMessage (**kwargs)
    Bases: google.protobuf.message.Message
    ByteSize ()
    CLIENTRESPONSE_FIELD_NUMBER = 1
    Clear ()
    ClearField (field_name)
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    FindInitializationErrors ()
        Finds required fields which are not initialized.
        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
            “foo.bar[5].baz”.
    static FromString (s)
    HasField (field_name)
    IsInitialized (errors=None)
        Checks if all required fields of a message are set.
        Args:
            errors: A list which, if provided, will be populated with the field paths of all missing required
                fields.
        Returns: True iff the specified message has all required fields set.
    ListFields ()
    MergeFrom (msg)
    MergeFromString (serialized)
    static RegisterExtension (extension_handle)
    SerializePartialToString ()
    SerializeToString ()
    SetInParent ()
        Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
    WhichOneof (oneof_name)
        Returns the name of the currently set field inside a oneof, or None.
```

clientResponse

Magic attribute generated for “clientResponse” proto field.

class pokerthproto.pokerth_pb2.**AuthServerChallengeMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SERVERCHALLENGE_FIELD_NUMBER = 1

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

serverChallenge

Magic attribute generated for “serverChallenge” proto field.

class pokerthproto.pokerth_pb2.**AuthServerVerificationMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SERVERVERIFICATION_FIELD_NUMBER = 1

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

serverVerification

Magic attribute generated for “serverVerification” proto field.

class pokerthproto.pokerth_pb2.**AvatarDataMessage** (**kwargs)
Bases: google.protobuf.message.Message

AVATARBLOCK_FIELD_NUMBER = 2

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REQUESTID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarBlock

Magic attribute generated for “avatarBlock” proto field.

requestId

Magic attribute generated for “requestId” proto field.

class pokerthproto.pokerth_pb2.**AvatarEndMessage** (**kwargs)
Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REQUESTID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

requestId

Magic attribute generated for “requestId” proto field.

class pokerthproto.pokerth_pb2.**AvatarHeaderMessage** (**kwargs)

Bases: google.protobuf.message.Message

AVATARSIZE_FIELD_NUMBER = 3

AVATARTYPE_FIELD_NUMBER = 2

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REQUESTID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarSize

Magic attribute generated for “avatarSize” proto field.

avatarType

Magic attribute generated for “avatarType” proto field.

requestId

Magic attribute generated for “requestId” proto field.

class `pokerthproto.pokerth_pb2.AvatarRequestMessage` (**kwargs)

Bases: `google.protobuf.message.Message`

AVATARHASH_FIELD_NUMBER = 2

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REQUESTID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarHash

Magic attribute generated for “avatarHash” proto field.

requestId

Magic attribute generated for “requestId” proto field.

class pokerthproto.pokerth_pb2.ChatMessage (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

CHATTEXT_FIELD_NUMBER = 4

CHATTYPE_FIELD_NUMBER = 3

ChatType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

chatText

Magic attribute generated for “chatText” proto field.

chatType

Magic attribute generated for “chatType” proto field.

chatTypeBot = 2

chatTypeBroadcast = 3

chatTypeGame = 1

chatTypeLobby = 0

chatTypePrivate = 4

gameId

Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**ChatRejectMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

CHATTEXT_FIELD_NUMBER = 1

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

chatText

Magic attribute generated for “chatText” proto field.

class pokerthproto.pokerth_pb2.**ChatRequestMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

CHATTEXT_FIELD_NUMBER = 3

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

TARGETGAMEID_FIELD_NUMBER = 1

TARGETPLAYERID_FIELD_NUMBER = 2

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

chatText

Magic attribute generated for “chatText” proto field.

targetGameId

Magic attribute generated for “targetGameId” proto field.

targetPlayerId

Magic attribute generated for “targetPlayerId” proto field.

class pokerthproto.pokerth_pb2.**DealFlopCardsMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FLOPCARD1_FIELD_NUMBER = 2

FLOPCARD2_FIELD_NUMBER = 3

FLOPCARD3_FIELD_NUMBER = 4

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

flopCard1

Magic attribute generated for “flopCard1” proto field.

flopCard2

Magic attribute generated for “flopCard2” proto field.

flopCard3

Magic attribute generated for “flopCard3” proto field.

gameId

Magic attribute generated for “gameId” proto field.

class pokerthproto.pokerth_pb2.**DealRiverCardMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

RIVERCARD_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

riverCard

Magic attribute generated for “riverCard” proto field.

class pokerthproto.pokerth_pb2.**DealTurnCardMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

```

ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.
    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        "foo.bar[5].baz".
static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.
Args:
    errors: A list which, if provided, will be populated with the field paths of all missing required
        fields.
Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
TURNCARD_FIELD_NUMBER = 2
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
gameId
    Magic attribute generated for "gameId" proto field.
turnCard
    Magic attribute generated for "turnCard" proto field.
class pokerthproto.pokerth_pb2.DialogMessage (**kwargs)
    Bases: google.protobuf.message.Message
ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.
    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        "foo.bar[5].baz".

```

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NOTIFICATIONTEXT_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

notificationText

Magic attribute generated for “notificationText” proto field.

class pokerthproto.pokerth_pb2.**EndKickPetitionMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NUMVOTESAGAINSTKICKING_FIELD_NUMBER = 3

NUMVOTESINFAVOUROFKICKING_FIELD_NUMBER = 4

PETITIONENDREASON_FIELD_NUMBER = 6

PETITIONID_FIELD_NUMBER = 2

PetitionEndReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

RESULTPLAYERKICKED_FIELD_NUMBER = 5

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

numVotesAgainstKicking

Magic attribute generated for “numVotesAgainstKicking” proto field.

numVotesInFavourOfKicking

Magic attribute generated for “numVotesInFavourOfKicking” proto field.

petitionEndEnoughVotes = 0

petitionEndPlayerLeft = 2

petitionEndReason

Magic attribute generated for “petitionEndReason” proto field.

petitionEndTimeout = 3

petitionEndTooFewPlayers = 1

petitionId

Magic attribute generated for “petitionId” proto field.

resultPlayerKicked

Magic attribute generated for “resultPlayerKicked” proto field.

class pokerthproto.pokerth_pb2.**EndOfGameMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WINNERPLAYERID_FIELD_NUMBER = 2

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

winnerPlayerId

Magic attribute generated for “winnerPlayerId” proto field.

class pokerthproto.pokerth_pb2.**EndOfHandHideCardsMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)
Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MONEYWON_FIELD_NUMBER = 3

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

PLAYERMONEY_FIELD_NUMBER = 4

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

gameId
Magic attribute generated for “gameId” proto field.

moneyWon
Magic attribute generated for “moneyWon” proto field.

playerId
Magic attribute generated for “playerId” proto field.

playerMoney
Magic attribute generated for “playerMoney” proto field.

class pokerthproto.pokerth_pb2.**EndOfHandShowCardsMessage** (**kwargs)
Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERRESULTS_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerResults

Magic attribute generated for “playerResults” proto field.

class pokerthproto.pokerth_pb2.**ErrorMessage** (**kwargs)
 Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

ERRORREASON_FIELD_NUMBER = 1

ErrorReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarTooLarge = 8

bannedFromServer = 12

blockedByServer = 13

errorReason

Magic attribute generated for “errorReason” proto field.

initAuthFailure = 3

initBlocked = 7

initInvalidPlayerName = 5

initPlayerNameInUse = 4

initServerFull = 2

initServerMaintenance = 6

initVersionNotSupported = 1

invalidPacket = 9

invalidState = 10

kickedFromServer = 11

reserved = 0

sessionTimeout = 14

class pokerthproto.pokerth_pb2.**GameAdminChangedMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NEWADMINPLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

newAdminPlayerId

Magic attribute generated for “newAdminPlayerId” proto field.

class pokerthproto.pokerth_pb2.**GameListAdminChangedMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NEWADMINPLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

newAdminPlayerId

Magic attribute generated for “newAdminPlayerId” proto field.

class pokerthproto.pokerth_pb2.**GameListNewMessage** (**kwargs)

Bases: google.protobuf.message.Message

ADMINPLAYERID_FIELD_NUMBER = 5

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMEINFO_FIELD_NUMBER = 6

GAMEMODE_FIELD_NUMBER = 2

HasField (*field_name*)

ISPRIVATE_FIELD_NUMBER = 3

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERIDS_FIELD_NUMBER = 4

static RegisterExtension (*extension_handle*)

SPECTATORIDS_FIELD_NUMBER = 7

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

adminPlayerId

Magic attribute generated for “adminPlayerId” proto field.

gameId

Magic attribute generated for “gameId” proto field.

gameInfo

Magic attribute generated for “gameInfo” proto field.

gameMode

Magic attribute generated for “gameMode” proto field.

isPrivate

Magic attribute generated for “isPrivate” proto field.

playerIds

Magic attribute generated for “playerIds” proto field.

spectatorIds

Magic attribute generated for “spectatorIds” proto field.

class pokerthproto.pokerth_pb2.**GameListPlayerJoinedMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**GameListPlayerLeftMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**GameListSpectatorJoinedMessage** (***kwargs*)

Bases: `google.protobuf.message.Message`

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.


```

ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
PLAYERID_FIELD_NUMBER = 2
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

gameId
    Magic attribute generated for “gameId” proto field.

playerId
    Magic attribute generated for “playerId” proto field.
class pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage (**kwargs)
    Bases: google.protobuf.message.Message

ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.

static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:

        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
PLAYERID_FIELD_NUMBER = 2
static RegisterExtension (extension_handle)

```

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**GameListUpdateMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMEMODE_FIELD_NUMBER = 2

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

gameMode

Magic attribute generated for “gameMode” proto field.

class pokerthproto.pokerth_pb2.**GamePlayerJoinedMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

ISGAMEADMIN_FIELD_NUMBER = 3

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

isGameAdmin

Magic attribute generated for “isGameAdmin” proto field.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**GamePlayerLeftMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMEPLAYERLEFTREASON_FIELD_NUMBER = 3

GamePlayerLeftReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

gamePlayerLeftReason

Magic attribute generated for “gamePlayerLeftReason” proto field.

leftError = 2

```

leftKicked = 1
leftOnRequest = 0
playerId
    Magic attribute generated for “playerId” proto field.
class pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage (**kwargs)
    Bases: google.protobuf.message.Message

ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.

static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:

        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
PLAYERID_FIELD_NUMBER = 2
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

gameId
    Magic attribute generated for “gameId” proto field.

playerId
    Magic attribute generated for “playerId” proto field.
class pokerthproto.pokerth_pb2.GameSpectatorLeftMessage (**kwargs)
    Bases: google.protobuf.message.Message

```

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMESPECTATORLEFTREASON_FIELD_NUMBER = 3

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for "gameId" proto field.

gameSpectatorLeftReason

Magic attribute generated for "gameSpectatorLeftReason" proto field.

playerId

Magic attribute generated for "playerId" proto field.

class pokerthproto.pokerth_pb2.**GameStartInitialMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

```

ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.
    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        "foo.bar[5].baz".
static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.
Args:
    errors: A list which, if provided, will be populated with the field paths of all missing required
        fields.
Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
PLAYERSEATS_FIELD_NUMBER = 3
static RegisterExtension (extension_handle)
STARTDEALERPLAYERID_FIELD_NUMBER = 2
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
gameId
    Magic attribute generated for "gameId" proto field.
playerSeats
    Magic attribute generated for "playerSeats" proto field.
startDealerPlayerId
    Magic attribute generated for "startDealerPlayerId" proto field.
class pokerthproto.pokerth_pb2.GameStartRejoinMessage (**kwargs)
    Bases: google.protobuf.message.Message
ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

```

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HANDNUM_FIELD_NUMBER = 3

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REJOINPLAYERDATA_FIELD_NUMBER = 4

static RegisterExtension (*extension_handle*)

class RejoinPlayerData (***kwargs*)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)


```

PLAYERID_FIELD_NUMBER = 1
PLAYERMONEY_FIELD_NUMBER = 2
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state
    change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

playerId
    Magic attribute generated for “playerId” proto field.

playerMoney
    Magic attribute generated for “playerMoney” proto field.

GameStartRejoinMessage.STARTDEALERPLAYERID_FIELD_NUMBER = 2
GameStartRejoinMessage.SerializePartialToString ()
GameStartRejoinMessage.SerializeToString ()
GameStartRejoinMessage.SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
GameStartRejoinMessage.WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

GameStartRejoinMessage.gameId
    Magic attribute generated for “gameId” proto field.

GameStartRejoinMessage.handNum
    Magic attribute generated for “handNum” proto field.

GameStartRejoinMessage.rejoinPlayerData
    Magic attribute generated for “rejoinPlayerData” proto field.

GameStartRejoinMessage.startDealerPlayerId
    Magic attribute generated for “startDealerPlayerId” proto field.

class pokerthproto.pokerth_pb2.HandStartMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()
    Clear ()
    ClearField (field_name)
    DEALERPLAYERID_FIELD_NUMBER = 6
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    ENCRYPTEDCARDS_FIELD_NUMBER = 3
    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.

```

```
static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.
    Args:
        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.
    Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
PLAINCARDS_FIELD_NUMBER = 2
class PlainCards (**kwargs)
    Bases: google.protobuf.message.Message
    ByteSize ()
    Clear ()
    ClearField (field_name)
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    FindInitializationErrors ()
        Finds required fields which are not initialized.
        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message,
            e.g. "foo.bar[5].baz".
    static FromString (s)
    HasField (field_name)
    IsInitialized (errors=None)
        Checks if all required fields of a message are set.
        Args:
            errors: A list which, if provided, will be populated with the field paths of all missing re-
                quired fields.
        Returns: True iff the specified message has all required fields set.
    ListFields ()
    MergeFrom (msg)
    MergeFromString (serialized)
    PLAINCARD1_FIELD_NUMBER = 1
    PLAINCARD2_FIELD_NUMBER = 2
    static RegisterExtension (extension_handle)
    SerializePartialToString ()
    SerializeToString ()
```

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

plainCard1
Magic attribute generated for “plainCard1” proto field.

plainCard2
Magic attribute generated for “plainCard2” proto field.

static `HandStartMessage.RegisterExtension` (*extension_handle*)

`HandStartMessage.SEATSTATES_FIELD_NUMBER = 5`

`HandStartMessage.SMALLBLIND_FIELD_NUMBER = 4`

`HandStartMessage.SerializePartialToString` ()

`HandStartMessage.SerializeToString` ()

`HandStartMessage.SetInParent` ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

`HandStartMessage.WhichOneof` (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

`HandStartMessage.dealerPlayerId`
Magic attribute generated for “dealerPlayerId” proto field.

`HandStartMessage.encryptedCards`
Magic attribute generated for “encryptedCards” proto field.

`HandStartMessage.gameId`
Magic attribute generated for “gameId” proto field.

`HandStartMessage.plainCards`
Magic attribute generated for “plainCards” proto field.

`HandStartMessage.seatStates`
Magic attribute generated for “seatStates” proto field.

`HandStartMessage.smallBlind`
Magic attribute generated for “smallBlind” proto field.

class `pokerthproto.pokerth_pb2.InitAckMessage` (**kwargs)
Bases: `google.protobuf.message.Message`

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static `FromString` (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REJOIN_GAMEID_FIELD_NUMBER = 4

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

YOURAVATARHASH_FIELD_NUMBER = 3

YOURPLAYERID_FIELD_NUMBER = 2

YOURSESSIONID_FIELD_NUMBER = 1

rejoinGameId

Magic attribute generated for “rejoinGameId” proto field.

yourAvatarHash

Magic attribute generated for “yourAvatarHash” proto field.

yourPlayerId

Magic attribute generated for “yourPlayerId” proto field.

yourSessionId

Magic attribute generated for “yourSessionId” proto field.

class pokerthproto.pokerth_pb2.**InitMessage** (**kwargs)

Bases: google.protobuf.message.Message

AUTHSERVERPASSWORD_FIELD_NUMBER = 4

AVATARHASH_FIELD_NUMBER = 8

BUILDID_FIELD_NUMBER = 2

ByteSize ()

CLIENTUSERDATA_FIELD_NUMBER = 7

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)**HasField** (*field_name*)**IsInitialized** (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

LOGIN_FIELD_NUMBER = 5

ListFields ()

LoginType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

MYLASTSESSIONID_FIELD_NUMBER = 3

MergeFrom (*msg*)**MergeFromString** (*serialized*)

NICKNAME_FIELD_NUMBER = 6

REQUESTEDVERSION_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)**SerializePartialToString** ()**SerializeToString** ()**SetInParent** ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

authServerPassword

Magic attribute generated for “authServerPassword” proto field.

authenticatedLogin = 1

avatarHash

Magic attribute generated for “avatarHash” proto field.

buildId

Magic attribute generated for “buildId” proto field.

clientUserData

Magic attribute generated for “clientUserData” proto field.

guestLogin = 0

login

Magic attribute generated for “login” proto field.

myLastSessionId

Magic attribute generated for “myLastSessionId” proto field.

nickName

Magic attribute generated for “nickName” proto field.

requestedVersion

Magic attribute generated for “requestedVersion” proto field.

unauthenticatedLogin = 2

class pokerthproto.pokerth_pb2.**InviteNotifyMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERIDBYWHOM_FIELD_NUMBER = 3

PLAYERIDWHO_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerIdByWhom

Magic attribute generated for “playerIdByWhom” proto field.

playerIdWho

Magic attribute generated for “playerIdWho” proto field.

```
class pokerthproto.pokerth_pb2.InvitePlayerToGameMessage (**kwargs)
```

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

```
class pokerthproto.pokerth_pb2.JoinExistingGameMessage (**kwargs)
```

Bases: google.protobuf.message.Message

AUTOLEAVE_FIELD_NUMBER = 3

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PASSWORD_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SPECTATEONLY_FIELD_NUMBER = 4

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

autoLeave

Magic attribute generated for “autoLeave” proto field.

gameId

Magic attribute generated for “gameId” proto field.

password

Magic attribute generated for “password” proto field.

spectateOnly

Magic attribute generated for “spectateOnly” proto field.

class pokerthproto.pokerth_pb2.**JoinGameAckMessage** (**kwargs)

Bases: google.protobuf.message.Message


```

AREYOUGAMEADMIN_FIELD_NUMBER = 2

ByteSize ()

Clear ()

ClearField (field_name)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
             "foo.bar[5].baz".

static FromString (s)

GAMEID_FIELD_NUMBER = 1

GAMEINFO_FIELD_NUMBER = 3

HasField (field_name)

IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:
        errors: A list which, if provided, will be populated with the field paths of all missing required
               fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (msg)

MergeFromString (serialized)

static RegisterExtension (extension_handle)

SPECTATEONLY_FIELD_NUMBER = 4

SerializePartialToString ()

SerializeToString ()

SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

areYouGameAdmin
    Magic attribute generated for "areYouGameAdmin" proto field.

gameId
    Magic attribute generated for "gameId" proto field.

gameInfo
    Magic attribute generated for "gameInfo" proto field.

spectateOnly
    Magic attribute generated for "spectateOnly" proto field.

class pokerthproto.pokerth_pb2.JoinGameFailedMessage (**kwargs)
    Bases: google.protobuf.message.Message

```

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

JOINGAMEFAILUREREASON_FIELD_NUMBER = 2

JoinGameFailureReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

badGameName = 8

gameId

Magic attribute generated for "gameId" proto field.

gameIsFull = 2

gameIsRunning = 3

gameNameInUse = 7

invalidGame = 1

invalidPassword = 4

invalidSettings = 9

ipAddressBlocked = 10

joinGameFailureReason

Magic attribute generated for “joinGameFailureReason” proto field.

noSpectatorsAllowed = 12

notAllowedAsGuest = 5

notInvited = 6

rejoinFailed = 11

class pokerthproto.pokerth_pb2.**JoinNewGameMessage** (**kwargs)

Bases: google.protobuf.message.Message

AUTOLEAVE_FIELD_NUMBER = 3

ByteSize ()

Clear ()

ClearField (field_name)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (s)

GAMEINFO_FIELD_NUMBER = 1

HasField (field_name)

IsInitialized (errors=None)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (msg)

MergeFromString (serialized)

PASSWORD_FIELD_NUMBER = 2

static RegisterExtension (extension_handle)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (oneof_name)

Returns the name of the currently set field inside a oneof, or None.

autoLeave

Magic attribute generated for “autoLeave” proto field.

gameInfo

Magic attribute generated for “gameInfo” proto field.

password

Magic attribute generated for “password” proto field.

class pokerthproto.pokerth_pb2.**KickPetitionUpdateMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NUMVOTESAGAINSTKICKING_FIELD_NUMBER = 3

NUMVOTESINFAVOUROFKICKING_FIELD_NUMBER = 4

NUMVOTESNEEDEDTOKICK_FIELD_NUMBER = 5

PETITIONID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

numVotesAgainstKicking

Magic attribute generated for “numVotesAgainstKicking” proto field.

numVotesInFavourOfKicking

Magic attribute generated for “numVotesInFavourOfKicking” proto field.

numVotesNeededToKick

Magic attribute generated for “numVotesNeededToKick” proto field.

petitionId

Magic attribute generated for “petitionId” proto field.

class pokerthproto.pokerth_pb2.**KickPlayerRequestMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId
 Magic attribute generated for “gameId” proto field.

playerId
 Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**LeaveGameRequestMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId
 Magic attribute generated for “gameId” proto field.

class pokerthproto.pokerth_pb2.**MyActionRequestMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMESTATE_FIELD_NUMBER = 3

HANDNUM_FIELD_NUMBER = 2

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MYACTION_FIELD_NUMBER = 4

MYRELATIVEBET_FIELD_NUMBER = 5

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

gameState

Magic attribute generated for “gameState” proto field.

handNum

Magic attribute generated for “handNum” proto field.

myAction

Magic attribute generated for “myAction” proto field.

myRelativeBet

Magic attribute generated for “myRelativeBet” proto field.

class pokerthproto.pokerth_pb2.**NetGameInfo** (***kwargs*)

Bases: google.protobuf.message.Message

`ALLOWSPECTATORS_FIELD_NUMBER = 15`

`ByteSize()`

`Clear()`

`ClearField(field_name)`

`DELAYBETWEENHANDS_FIELD_NUMBER = 10`

`DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>`

`ENDRAISEMODE_FIELD_NUMBER = 7`

`ENDRAISESMALLBLINDVALUE_FIELD_NUMBER = 8`

`EndRaiseMode = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>`

`FIRSTSMALLBLIND_FIELD_NUMBER = 12`

`FindInitializationErrors()`
Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".

`static FromString(s)`

`GAMENAME_FIELD_NUMBER = 1`

`HasField(field_name)`

`IsInitialized(errors=None)`
Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

`ListFields()`

`MANUALBLINDS_FIELD_NUMBER = 14`

`MAXNUMPLAYERS_FIELD_NUMBER = 3`

`MergeFrom(msg)`

`MergeFromString(serialized)`

`NETGAMETYPE_FIELD_NUMBER = 2`

`NetGameType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>`

`PLAYERACTIONTIMEOUT_FIELD_NUMBER = 11`

`PROPOSEDGUI SPEED_FIELD_NUMBER = 9`

`RAISEEVERYHANDS_FIELD_NUMBER = 5`

`RAISEEVERYMINUTES_FIELD_NUMBER = 6`

`RAISEINTERVALMODE_FIELD_NUMBER = 4`

`RaiseIntervalMode = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>`

`static RegisterExtension(extension_handle)`

STARTMONEY_FIELD_NUMBER = 13

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

allowSpectators

Magic attribute generated for “allowSpectators” proto field.

delayBetweenHands

Magic attribute generated for “delayBetweenHands” proto field.

doubleBlinds = 1

endRaiseMode

Magic attribute generated for “endRaiseMode” proto field.

endRaiseSmallBlindValue

Magic attribute generated for “endRaiseSmallBlindValue” proto field.

firstSmallBlind

Magic attribute generated for “firstSmallBlind” proto field.

gameName

Magic attribute generated for “gameName” proto field.

inviteOnlyGame = 3

keepLastBlind = 3

manualBlinds

Magic attribute generated for “manualBlinds” proto field.

maxNumPlayers

Magic attribute generated for “maxNumPlayers” proto field.

netGameType

Magic attribute generated for “netGameType” proto field.

normalGame = 1

playerActionTimeout

Magic attribute generated for “playerActionTimeout” proto field.

proposedGuiSpeed

Magic attribute generated for “proposedGuiSpeed” proto field.

raiseByEndValue = 2

raiseEveryHands

Magic attribute generated for “raiseEveryHands” proto field.

raiseEveryMinutes

Magic attribute generated for “raiseEveryMinutes” proto field.

raiseIntervalMode

Magic attribute generated for “raiseIntervalMode” proto field.

raiseOnHandNum = 1

raiseOnMinutes = 2

rankingGame = 4

registeredOnlyGame = 2

startMoney

Magic attribute generated for “startMoney” proto field.

class pokerthproto.pokerth_pb2.**PlayerIdChangedMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NEWPLAYERID_FIELD_NUMBER = 2

OLDPLAYERID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

newPlayerId

Magic attribute generated for “newPlayerId” proto field.

oldPlayerId

Magic attribute generated for “oldPlayerId” proto field.

```

class pokerthproto.pokerth_pb2.PlayerInfoReplyMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
            "foo.bar[5].baz".

    static FromString (s)

    HasField (field_name)

    IsInitialized (errors=None)
        Checks if all required fields of a message are set.

    Args:
        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.

    Returns: True iff the specified message has all required fields set.

    ListFields ()

    MergeFrom (msg)

    MergeFromString (serialized)

    PLAYERID_FIELD_NUMBER = 1

    PLAYERINFODATA_FIELD_NUMBER = 2

class PlayerInfoData (**kwargs)
    Bases: google.protobuf.message.Message

    AVATARDATA_FIELD_NUMBER = 5

class AvatarData (**kwargs)
    Bases: google.protobuf.message.Message

    AVATARHASH_FIELD_NUMBER = 2

    AVATARTYPE_FIELD_NUMBER = 1

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level mes-
            sage, e.g. "foo.bar[5].baz".

    static FromString (s)

```

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarHash

Magic attribute generated for “avatarHash” proto field.

avatarType

Magic attribute generated for “avatarType” proto field.

`PlayerInfoReplyMessage.PlayerInfoData`.**ByteSize** ()

`PlayerInfoReplyMessage.PlayerInfoData`.**COUNTRYCODE_FIELD_NUMBER** = 4

`PlayerInfoReplyMessage.PlayerInfoData`.**Clear** ()

`PlayerInfoReplyMessage.PlayerInfoData`.**ClearField** (*field_name*)

`PlayerInfoReplyMessage.PlayerInfoData`.**DESCRIPTOR** = <google.protobuf.descriptor.Descriptor object>

`PlayerInfoReplyMessage.PlayerInfoData`.**FindInitializationErrors** ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static `PlayerInfoReplyMessage.PlayerInfoData`.**FromString** (*s*)

`PlayerInfoReplyMessage.PlayerInfoData`.**HasField** (*field_name*)

`PlayerInfoReplyMessage.PlayerInfoData`.**ISHUMAN_FIELD_NUMBER** = 2

`PlayerInfoReplyMessage.PlayerInfoData`.**IsInitialized** (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

`PlayerInfoReplyMessage.PlayerInfoData`.**ListFields** ()

`PlayerInfoReplyMessage.PlayerInfoData`.**MergeFrom** (*msg*)

```

PlayerInfoReplyMessage.PlayerInfoData.MergeFromString (serialized)
PlayerInfoReplyMessage.PlayerInfoData.PLAYERNAME_FIELD_NUMBER = 1
PlayerInfoReplyMessage.PlayerInfoData.PLAYERRIGHTS_FIELD_NUMBER = 3
static PlayerInfoReplyMessage.PlayerInfoData.RegisterExtension (extension_handle)
PlayerInfoReplyMessage.PlayerInfoData.SerializePartialToString ()
PlayerInfoReplyMessage.PlayerInfoData.SerializeToString ()
PlayerInfoReplyMessage.PlayerInfoData.SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state
    change.
PlayerInfoReplyMessage.PlayerInfoData.WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
PlayerInfoReplyMessage.PlayerInfoData.avatarData
    Magic attribute generated for “avatarData” proto field.
PlayerInfoReplyMessage.PlayerInfoData.countryCode
    Magic attribute generated for “countryCode” proto field.
PlayerInfoReplyMessage.PlayerInfoData.isHuman
    Magic attribute generated for “isHuman” proto field.
PlayerInfoReplyMessage.PlayerInfoData.playerName
    Magic attribute generated for “playerName” proto field.
PlayerInfoReplyMessage.PlayerInfoData.playerRights
    Magic attribute generated for “playerRights” proto field.
static PlayerInfoReplyMessage.RegisterExtension (extension_handle)
PlayerInfoReplyMessage.SerializePartialToString ()
PlayerInfoReplyMessage.SerializeToString ()
PlayerInfoReplyMessage.SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
PlayerInfoReplyMessage.WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
PlayerInfoReplyMessage.playerId
    Magic attribute generated for “playerId” proto field.
PlayerInfoReplyMessage.playerInfoData
    Magic attribute generated for “playerInfoData” proto field.
class pokerthproto.pokerth_pb2.PlayerInfoRequestMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()
    Clear ()
    ClearField (field_name)
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    FindInitializationErrors ()
        Finds required fields which are not initialized.

```

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

playerId

Magic attribute generated for “playerId” proto field.

class pokerthproto.pokerth_pb2.**PlayerListMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 1

PLAYERLISTNOTIFICATION_FIELD_NUMBER = 2

PlayerListNotification = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

playerId

Magic attribute generated for “playerId” proto field.

playerListLeft = 1

playerListNew = 0

playerListNotification

Magic attribute generated for “playerListNotification” proto field.

class pokerthproto.pokerth_pb2.**PlayerResult** (**kwargs)

Bases: google.protobuf.message.Message

BESTHANDPOSITION_FIELD_NUMBER = 4

ByteSize ()

CARDSVALUE_FIELD_NUMBER = 7

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MONEYWON_FIELD_NUMBER = 5

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 1

PLAYERMONEY_FIELD_NUMBER = 6

RESULTCARD1_FIELD_NUMBER = 2

RESULTCARD2_FIELD_NUMBER = 3

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

bestHandPosition

Magic attribute generated for “bestHandPosition” proto field.

cardsValue

Magic attribute generated for “cardsValue” proto field.

moneyWon

Magic attribute generated for “moneyWon” proto field.

playerId

Magic attribute generated for “playerId” proto field.

playerMoney

Magic attribute generated for “playerMoney” proto field.

resultCard1

Magic attribute generated for “resultCard1” proto field.

resultCard2

Magic attribute generated for “resultCard2” proto field.

class pokerthproto.pokerth_pb2.**PlayersActionDoneMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMESTATE_FIELD_NUMBER = 3

HIGHESTSET_FIELD_NUMBER = 7

HasField (*field_name*)

IsInitialized (*errors=None*)
Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MINIMUMRAISE_FIELD_NUMBER = 8

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERACTION_FIELD_NUMBER = 4

PLAYERID_FIELD_NUMBER = 2

PLAYERMONEY_FIELD_NUMBER = 6

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

TOTALPLAYERBET_FIELD_NUMBER = 5

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

gameId
Magic attribute generated for “gameId” proto field.

gameState
Magic attribute generated for “gameState” proto field.

highestSet
Magic attribute generated for “highestSet” proto field.

minimumRaise
Magic attribute generated for “minimumRaise” proto field.

playerAction
Magic attribute generated for “playerAction” proto field.

playerId
Magic attribute generated for “playerId” proto field.

playerMoney
Magic attribute generated for “playerMoney” proto field.

totalPlayerBet

Magic attribute generated for “totalPlayerBet” proto field.

class pokerthproto.pokerth_pb2.**PlayersTurnMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

GAMESTATE_FIELD_NUMBER = 3

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId

Magic attribute generated for “gameId” proto field.

gameState

Magic attribute generated for “gameState” proto field.

playerId

Magic attribute generated for “playerId” proto field.

```

class pokerthproto.pokerth_pb2.PokerTHMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ADMINBANPLAYERACKMESSAGE_FIELD_NUMBER = 78
    ADMINBANPLAYERMESSAGE_FIELD_NUMBER = 77
    ADMINREMOVEGAMEACKMESSAGE_FIELD_NUMBER = 76
    ADMINREMOVEGAMEMESSAGE_FIELD_NUMBER = 75
    AFTERHANDSHOWCARDSMESSAGE_FIELD_NUMBER = 53
    ALLINSHOWCARDSMESSAGE_FIELD_NUMBER = 49
    ANNOUNCEMESSAGE_FIELD_NUMBER = 2
    ASKKICKDENIEDMESSAGE_FIELD_NUMBER = 57
    ASKKICKPLAYERMESSAGE_FIELD_NUMBER = 56
    AUTHCLIENTRESPONSEMESSAGE_FIELD_NUMBER = 5
    AUTHSERVERCHALLENGEMESSAGE_FIELD_NUMBER = 4
    AUTHSERVERVERIFICATIONMESSAGE_FIELD_NUMBER = 6
    AVATARDATAMESSAGE_FIELD_NUMBER = 10
    AVATARENDMESSAGE_FIELD_NUMBER = 11
    AVATARHEADERMESSAGE_FIELD_NUMBER = 9
    AVATARREQUESTMESSAGE_FIELD_NUMBER = 8
    ByteSize ()
    CHATMESSAGE_FIELD_NUMBER = 65
    CHATREJECTMESSAGE_FIELD_NUMBER = 66
    CHATREQUESTMESSAGE_FIELD_NUMBER = 64
    Clear ()
    ClearField (field_name)
    DEALFLOPCARDSMESSAGE_FIELD_NUMBER = 46
    DEALRIVERCARDMESSAGE_FIELD_NUMBER = 48
    DEALTURNCARDMESSAGE_FIELD_NUMBER = 47
    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
    DIALOGMESSAGE_FIELD_NUMBER = 67
    ENDKICKPETITIONMESSAGE_FIELD_NUMBER = 62
    ENDOFGAMEMESSAGE_FIELD_NUMBER = 54
    ENDOFHANDHIDECARDSMESSAGE_FIELD_NUMBER = 51
    ENDOFHANDSHOWCARDSMESSAGE_FIELD_NUMBER = 50
    ERRORMESSAGE_FIELD_NUMBER = 74
    FindInitializationErrors ()
        Finds required fields which are not initialized.

```

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

`static FromString (s)`

`GAMEADMINCHANGEDMESSAGE_FIELD_NUMBER = 29`

`GAMELISTADMINCHANGEDMESSAGE_FIELD_NUMBER = 18`

`GAMELISTNEWMESSAGE_FIELD_NUMBER = 14`

`GAMELISTPLAYERJOINEDMESSAGE_FIELD_NUMBER = 16`

`GAMELISTPLAYERLEFTMESSAGE_FIELD_NUMBER = 17`

`GAMELISTSPECTATORJOINEDMESSAGE_FIELD_NUMBER = 79`

`GAMELISTSPECTATORLEFTMESSAGE_FIELD_NUMBER = 80`

`GAMELISTUPDATEMESSAGE_FIELD_NUMBER = 15`

`GAMEPLAYERJOINEDMESSAGE_FIELD_NUMBER = 27`

`GAMEPLAYERLEFTMESSAGE_FIELD_NUMBER = 28`

`GAMESPECTATORJOINEDMESSAGE_FIELD_NUMBER = 81`

`GAMESPECTATORLEFTMESSAGE_FIELD_NUMBER = 82`

`GAMESTARTINITIALMESSAGE_FIELD_NUMBER = 39`

`GAMESTARTREJOINMESSAGE_FIELD_NUMBER = 40`

`HANDSTARTMESSAGE_FIELD_NUMBER = 41`

`HasField (field_name)`

`INITACKMESSAGE_FIELD_NUMBER = 7`

`INITMESSAGE_FIELD_NUMBER = 3`

`INVITENOTIFYMESSAGE_FIELD_NUMBER = 34`

`INVITEPLAYERTOGAMEMESSAGE_FIELD_NUMBER = 33`

`IsInitialized (errors=None)`

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

`JOINEXISTINGGAMEMESSAGE_FIELD_NUMBER = 22`

`JOINGAMEACKMESSAGE_FIELD_NUMBER = 25`

`JOINGAMEFAILEDMESSAGE_FIELD_NUMBER = 26`

`JOINNEWGAMEMESSAGE_FIELD_NUMBER = 23`

`KICKPETITIONUPDATEMESSAGE_FIELD_NUMBER = 61`

`KICKPLAYERREQUESTMESSAGE_FIELD_NUMBER = 31`

`LEAVEGAMEREQUESTMESSAGE_FIELD_NUMBER = 32`

`ListFields ()`

```

MESSAGE_TYPE_FIELD_NUMBER = 1
MYACTIONREQUESTMESSAGE_FIELD_NUMBER = 43
MergeFrom(msg)
MergeFromString(serialized)
PLAYERIDCHANGEDMESSAGE_FIELD_NUMBER = 55
PLAYERINFOREPLYMESSAGE_FIELD_NUMBER = 20
PLAYERINFOREQUESTMESSAGE_FIELD_NUMBER = 19
PLAYERLISTMESSAGE_FIELD_NUMBER = 13
PLAYERSACTIONDONEMESSAGE_FIELD_NUMBER = 45
PLAYERSTURNMESSAGE_FIELD_NUMBER = 42
PokerTHMessageType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>
REJECTGAMEINVITATIONMESSAGE_FIELD_NUMBER = 35
REJECTINVNOTIFYMESSAGE_FIELD_NUMBER = 36
REJOINEXISTINGGAMEMESSAGE_FIELD_NUMBER = 24
REMOVEDFROMGAMEMESSAGE_FIELD_NUMBER = 30
REPORTAVATARACKMESSAGE_FIELD_NUMBER = 71
REPORTAVATARMESSAGE_FIELD_NUMBER = 70
REPORTGAMEACKMESSAGE_FIELD_NUMBER = 73
REPORTGAMEMESSAGE_FIELD_NUMBER = 72
RESETTIMEOUTMESSAGE_FIELD_NUMBER = 69
static RegisterExtension(extension_handle)
SHOWMYCARDSREQUESTMESSAGE_FIELD_NUMBER = 52
STARTEVENTACKMESSAGE_FIELD_NUMBER = 38
STARTEVENTMESSAGE_FIELD_NUMBER = 37
STARTKICKPETITIONMESSAGE_FIELD_NUMBER = 58
STATISTICSMESSAGE_FIELD_NUMBER = 63
SUBSCRIPTIONREQUESTMESSAGE_FIELD_NUMBER = 21
SerializePartialToString()
SerializeToString()
SetInParent()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
TIMEOUTWARNINGMESSAGE_FIELD_NUMBER = 68
Type_AdminBanPlayerAckMessage = 77
Type_AdminBanPlayerMessage = 76
Type_AdminRemoveGameAckMessage = 75
Type_AdminRemoveGameMessage = 74

```

Type_AfterHandShowCardsMessage = 52
Type_AllInShowCardsMessage = 48
Type_AnnounceMessage = 1
Type_AskKickDeniedMessage = 56
Type_AskKickPlayerMessage = 55
Type_AuthClientResponseMessage = 4
Type_AuthServerChallengeMessage = 3
Type_AuthServerVerificationMessage = 5
Type_AvatarDataMessage = 9
Type_AvatarEndMessage = 10
Type_AvatarHeaderMessage = 8
Type_AvatarRequestMessage = 7
Type_ChatMessage = 64
Type_ChatRejectMessage = 65
Type_ChatRequestMessage = 63
Type_DealFlopCardsMessage = 45
Type_DealRiverCardMessage = 47
Type_DealTurnCardMessage = 46
Type_DialogMessage = 66
Type_EndKickPetitionMessage = 61
Type_EndOfGameMessage = 53
Type_EndOfHandHideCardsMessage = 50
Type_EndOfHandShowCardsMessage = 49
Type_ErrorMessage = 73
Type_GameAdminChangedMessage = 28
Type_GameListAdminChangedMessage = 17
Type_GameListNewMessage = 13
Type_GameListPlayerJoinedMessage = 15
Type_GameListPlayerLeftMessage = 16
Type_GameListSpectatorJoinedMessage = 78
Type_GameListSpectatorLeftMessage = 79
Type_GameListUpdateMessage = 14
Type_GamePlayerJoinedMessage = 26
Type_GamePlayerLeftMessage = 27
Type_GameSpectatorJoinedMessage = 80
Type_GameSpectatorLeftMessage = 81

Type_GameStartInitialMessage = 38
Type_GameStartRejoinMessage = 39
Type_HandStartMessage = 40
Type_InitAckMessage = 6
Type_InitMessage = 2
Type_InviteNotifyMessage = 33
Type_InvitePlayerToGameMessage = 32
Type_JoinExistingGameMessage = 21
Type_JoinGameAckMessage = 24
Type_JoinGameFailedMessage = 25
Type_JoinNewGameMessage = 22
Type_KickPetitionUpdateMessage = 60
Type_KickPlayerRequestMessage = 30
Type_LeaveGameRequestMessage = 31
Type_MyActionRequestMessage = 42
Type_PlayerIdChangedMessage = 54
Type_PlayerInfoReplyMessage = 19
Type_PlayerInfoRequestMessage = 18
Type_PlayerListMessage = 12
Type_PlayersActionDoneMessage = 44
Type_PlayersTurnMessage = 41
Type_RejectGameInvitationMessage = 34
Type_RejectInvNotifyMessage = 35
Type_RejoinExistingGameMessage = 23
Type_RemovedFromGameMessage = 29
Type_ReportAvatarAckMessage = 70
Type_ReportAvatarMessage = 69
Type_ReportGameAckMessage = 72
Type_ReportGameMessage = 71
Type_ResetTimeoutMessage = 68
Type_ShowMyCardsRequestMessage = 51
Type_StartEventAckMessage = 37
Type_StartEventMessage = 36
Type_StartKickPetitionMessage = 57
Type_StatisticsMessage = 62
Type_SubscriptionRequestMessage = 20

Type_TimeoutWarningMessage = 67

Type_UnknownAvatarMessage = 11

Type_VoteKickReplyMessage = 59

Type_VoteKickRequestMessage = 58

Type_YourActionRejectedMessage = 43

UNKNOWNAVATARMESSAGE_FIELD_NUMBER = 12

VOTEKICKREPLYMESSAGE_FIELD_NUMBER = 60

VOTEKICKREQUESTMESSAGE_FIELD_NUMBER = 59

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

YOURACTIONREJECTEDMESSAGE_FIELD_NUMBER = 44

adminBanPlayerAckMessage

Magic attribute generated for “adminBanPlayerAckMessage” proto field.

adminBanPlayerMessage

Magic attribute generated for “adminBanPlayerMessage” proto field.

adminRemoveGameAckMessage

Magic attribute generated for “adminRemoveGameAckMessage” proto field.

adminRemoveGameMessage

Magic attribute generated for “adminRemoveGameMessage” proto field.

afterHandShowCardsMessage

Magic attribute generated for “afterHandShowCardsMessage” proto field.

allInShowCardsMessage

Magic attribute generated for “allInShowCardsMessage” proto field.

announceMessage

Magic attribute generated for “announceMessage” proto field.

askKickDeniedMessage

Magic attribute generated for “askKickDeniedMessage” proto field.

askKickPlayerMessage

Magic attribute generated for “askKickPlayerMessage” proto field.

authClientResponseMessage

Magic attribute generated for “authClientResponseMessage” proto field.

authServerChallengeMessage

Magic attribute generated for “authServerChallengeMessage” proto field.

authServerVerificationMessage

Magic attribute generated for “authServerVerificationMessage” proto field.

avatarDataMessage

Magic attribute generated for “avatarDataMessage” proto field.

avatarEndMessage

Magic attribute generated for “avatarEndMessage” proto field.

avatarHeaderMessage

Magic attribute generated for “avatarHeaderMessage” proto field.

avatarRequestMessage

Magic attribute generated for “avatarRequestMessage” proto field.

chatMessage

Magic attribute generated for “chatMessage” proto field.

chatRejectMessage

Magic attribute generated for “chatRejectMessage” proto field.

chatRequestMessage

Magic attribute generated for “chatRequestMessage” proto field.

dealFlopCardsMessage

Magic attribute generated for “dealFlopCardsMessage” proto field.

dealRiverCardMessage

Magic attribute generated for “dealRiverCardMessage” proto field.

dealTurnCardMessage

Magic attribute generated for “dealTurnCardMessage” proto field.

dialogMessage

Magic attribute generated for “dialogMessage” proto field.

endKickPetitionMessage

Magic attribute generated for “endKickPetitionMessage” proto field.

endOfGameMessage

Magic attribute generated for “endOfGameMessage” proto field.

endOfHandHideCardsMessage

Magic attribute generated for “endOfHandHideCardsMessage” proto field.

endOfHandShowCardsMessage

Magic attribute generated for “endOfHandShowCardsMessage” proto field.

errorMessage

Magic attribute generated for “errorMessage” proto field.

gameAdminChangedMessage

Magic attribute generated for “gameAdminChangedMessage” proto field.

gameListAdminChangedMessage

Magic attribute generated for “gameListAdminChangedMessage” proto field.

gameListNewMessage

Magic attribute generated for “gameListNewMessage” proto field.

gameListPlayerJoinedMessage

Magic attribute generated for “gameListPlayerJoinedMessage” proto field.

gameListPlayerLeftMessage

Magic attribute generated for “gameListPlayerLeftMessage” proto field.

gameListSpectatorJoinedMessage

Magic attribute generated for “gameListSpectatorJoinedMessage” proto field.

gameListSpectatorLeftMessage

Magic attribute generated for “gameListSpectatorLeftMessage” proto field.

gameListUpdateMessage

Magic attribute generated for “gameListUpdateMessage” proto field.

gamePlayerJoinedMessage

Magic attribute generated for “gamePlayerJoinedMessage” proto field.

gamePlayerLeftMessage

Magic attribute generated for “gamePlayerLeftMessage” proto field.

gameSpectatorJoinedMessage

Magic attribute generated for “gameSpectatorJoinedMessage” proto field.

gameSpectatorLeftMessage

Magic attribute generated for “gameSpectatorLeftMessage” proto field.

gameStartInitialMessage

Magic attribute generated for “gameStartInitialMessage” proto field.

gameStartRejoinMessage

Magic attribute generated for “gameStartRejoinMessage” proto field.

handStartMessage

Magic attribute generated for “handStartMessage” proto field.

initAckMessage

Magic attribute generated for “initAckMessage” proto field.

initMessage

Magic attribute generated for “initMessage” proto field.

inviteNotifyMessage

Magic attribute generated for “inviteNotifyMessage” proto field.

invitePlayerToGameMessage

Magic attribute generated for “invitePlayerToGameMessage” proto field.

joinExistingGameMessage

Magic attribute generated for “joinExistingGameMessage” proto field.

joinGameAckMessage

Magic attribute generated for “joinGameAckMessage” proto field.

joinGameFailedMessage

Magic attribute generated for “joinGameFailedMessage” proto field.

joinNewGameMessage

Magic attribute generated for “joinNewGameMessage” proto field.

kickPetitionUpdateMessage

Magic attribute generated for “kickPetitionUpdateMessage” proto field.

kickPlayerRequestMessage

Magic attribute generated for “kickPlayerRequestMessage” proto field.

leaveGameRequestMessage

Magic attribute generated for “leaveGameRequestMessage” proto field.

messageType

Magic attribute generated for “messageType” proto field.

myActionRequestMessage

Magic attribute generated for “myActionRequestMessage” proto field.

playerIdChangedMessage

Magic attribute generated for “playerIdChangedMessage” proto field.

playerInfoReplyMessage

Magic attribute generated for “playerInfoReplyMessage” proto field.

playerInfoRequestMessage

Magic attribute generated for “playerInfoRequestMessage” proto field.

playerListMessage

Magic attribute generated for “playerListMessage” proto field.

playersActionDoneMessage

Magic attribute generated for “playersActionDoneMessage” proto field.

playersTurnMessage

Magic attribute generated for “playersTurnMessage” proto field.

rejectGameInvitationMessage

Magic attribute generated for “rejectGameInvitationMessage” proto field.

rejectInvNotifyMessage

Magic attribute generated for “rejectInvNotifyMessage” proto field.

rejoinExistingGameMessage

Magic attribute generated for “rejoinExistingGameMessage” proto field.

removedFromGameMessage

Magic attribute generated for “removedFromGameMessage” proto field.

reportAvatarAckMessage

Magic attribute generated for “reportAvatarAckMessage” proto field.

reportAvatarMessage

Magic attribute generated for “reportAvatarMessage” proto field.

reportGameAckMessage

Magic attribute generated for “reportGameAckMessage” proto field.

reportGameMessage

Magic attribute generated for “reportGameMessage” proto field.

resetTimeoutMessage

Magic attribute generated for “resetTimeoutMessage” proto field.

showMyCardsRequestMessage

Magic attribute generated for “showMyCardsRequestMessage” proto field.

startEventAckMessage

Magic attribute generated for “startEventAckMessage” proto field.

startEventMessage

Magic attribute generated for “startEventMessage” proto field.

startKickPetitionMessage

Magic attribute generated for “startKickPetitionMessage” proto field.

statisticsMessage

Magic attribute generated for “statisticsMessage” proto field.

subscriptionRequestMessage

Magic attribute generated for “subscriptionRequestMessage” proto field.

timeoutWarningMessage

Magic attribute generated for “timeoutWarningMessage” proto field.

unknownAvatarMessage

Magic attribute generated for “unknownAvatarMessage” proto field.

voteKickReplyMessage

Magic attribute generated for “voteKickReplyMessage” proto field.

voteKickRequestMessage

Magic attribute generated for “voteKickRequestMessage” proto field.

yourActionRejectedMessage

Magic attribute generated for “yourActionRejectedMessage” proto field.

class pokerthproto.pokerth_pb2.**RejectGameInvitationMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MYREJECTREASON_FIELD_NUMBER = 2

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

RejectGameInvReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

gameId
Magic attribute generated for “gameId” proto field.

myRejectReason
Magic attribute generated for “myRejectReason” proto field.

rejectReasonBusy = 1

rejectReasonNo = 0

class pokerthproto.pokerth_pb2.**RejectInvNotifyMessage** (**kwargs)
Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
Finds required fields which are not initialized.
Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)
Checks if all required fields of a message are set.
Args:
errors: A list which, if provided, will be populated with the field paths of all missing required fields.
Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PLAYERID_FIELD_NUMBER = 2

PLAYERREJECTREASON_FIELD_NUMBER = 3

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

gameId
Magic attribute generated for “gameId” proto field.

playerId

Magic attribute generated for “playerId” proto field.

playerRejectReason

Magic attribute generated for “playerRejectReason” proto field.

class pokerthproto.pokerth_pb2.**RejoinExistingGameMessage** (**kwargs)

Bases: google.protobuf.message.Message

AUTOLEAVE_FIELD_NUMBER = 2

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

autoLeave

Magic attribute generated for “autoLeave” proto field.

gameId

Magic attribute generated for “gameId” proto field.

class pokerthproto.pokerth_pb2.**RemovedFromGameMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()
Clear ()
ClearField (*field_name*)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
 Finds required fields which are not initialized.
Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".
static FromString (*s*)
GAMEID_FIELD_NUMBER = 1
HasField (*field_name*)
IsInitialized (*errors=None*)
 Checks if all required fields of a message are set.
Args:
errors: A list which, if provided, will be populated with the field paths of all missing required fields.
Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (*msg*)
MergeFromString (*serialized*)
REMOVEDFROMGAMEREASON_FIELD_NUMBER = 2
static RegisterExtension (*extension_handle*)
RemovedFromGameReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>
SerializePartialToString ()
SerializeToString ()
SetInParent ()
 Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (*oneof_name*)
 Returns the name of the currently set field inside a oneof, or None.
gameClosed = 6
gameId
 Magic attribute generated for "gameId" proto field.
gameIsFull = 2
gameIsRunning = 3
gameTimeout = 4
kickedFromGame = 1
removedFromGameReason
 Magic attribute generated for "removedFromGameReason" proto field.
removedOnRequest = 0

```
removedStartFailed = 5
```

class pokerthproto.pokerth_pb2.**ReportAvatarAckMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REPORTAVATARRESULT_FIELD_NUMBER = 2

REPORTEDPLAYERID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

ReportAvatarResult = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

avatarReportAccepted = 0

avatarReportDuplicate = 1

avatarReportInvalid = 2

reportAvatarResult

Magic attribute generated for “reportAvatarResult” proto field.

reportedPlayerId

Magic attribute generated for “reportedPlayerId” proto field.

```

class pokerthproto.pokerth_pb2.ReportAvatarMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
            "foo.bar[5].baz".

    static FromString (s)

    HasField (field_name)

    IsInitialized (errors=None)
        Checks if all required fields of a message are set.

        Args:
            errors: A list which, if provided, will be populated with the field paths of all missing required
                fields.

        Returns: True iff the specified message has all required fields set.

    ListFields ()

    MergeFrom (msg)

    MergeFromString (serialized)

    REPORTEDAVATARHASH_FIELD_NUMBER = 2

    REPORTEDPLAYERID_FIELD_NUMBER = 1

    static RegisterExtension (extension_handle)

    SerializePartialToString ()

    SerializeToString ()

    SetInParent ()
        Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

    WhichOneof (oneof_name)
        Returns the name of the currently set field inside a oneof, or None.

    reportedAvatarHash
        Magic attribute generated for "reportedAvatarHash" proto field.

    reportedPlayerId
        Magic attribute generated for "reportedPlayerId" proto field.

class pokerthproto.pokerth_pb2.ReportGameAckMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

```

```

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        "foo.bar[5].baz".

static FromString (s)

HasField (field_name)

IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:
        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (msg)

MergeFromString (serialized)

REPORTEDGAMEID_FIELD_NUMBER = 1

REPORTGAMERESULT_FIELD_NUMBER = 2

static RegisterExtension (extension_handle)

ReportGameResult = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

SerializePartialToString ()

SerializeToString ()

SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

gameReportAccepted = 0

gameReportDuplicate = 1

gameReportInvalid = 2

reportGameResult
    Magic attribute generated for "reportGameResult" proto field.

reportedGameId
    Magic attribute generated for "reportedGameId" proto field.

class pokerthproto.pokerth_pb2.ReportGameMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

```

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)**HasField** (*field_name*)**IsInitialized** (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()**MergeFrom** (*msg*)**MergeFromString** (*serialized*)**REPORTEDGAMEID_FIELD_NUMBER = 1****static RegisterExtension** (*extension_handle*)**SerializePartialToString** ()**SerializeToString** ()**SetInParent** ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

reportedGameId

Magic attribute generated for “reportedGameId” proto field.

```
class pokerthproto.pokerth_pb2.ResetTimeoutMessage (**kwargs)
    Bases: google.protobuf.message.Message
```

ByteSize ()**Clear** ()**ClearField** (*field_name*)**DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>****FindInitializationErrors** ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)**HasField** (*field_name*)**IsInitialized** (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

class pokerthproto.pokerth_pb2.**ShowMyCardsRequestMessage** (**kwargs)
 Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. "foo.bar[5].baz".

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

```

WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

class pokerthproto.pokerth_pb2.StartEventAckMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

    FindInitializationErrors ()
        Finds required fields which are not initialized.

        Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
            "foo.bar[5].baz".

    static FromString (s)

    GAMEID_FIELD_NUMBER = 1

    HasField (field_name)

    IsInitialized (errors=None)
        Checks if all required fields of a message are set.

        Args:

            errors: A list which, if provided, will be populated with the field paths of all missing required
                fields.

        Returns: True iff the specified message has all required fields set.

    ListFields ()

    MergeFrom (msg)

    MergeFromString (serialized)

    static RegisterExtension (extension_handle)

    SerializePartialToString ()

    SerializeToString ()

    SetInParent ()
        Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.

    WhichOneof (oneof_name)
        Returns the name of the currently set field inside a oneof, or None.

    gameId
        Magic attribute generated for "gameId" proto field.

class pokerthproto.pokerth_pb2.StartEventMessage (**kwargs)
    Bases: google.protobuf.message.Message

    ByteSize ()

    Clear ()

    ClearField (field_name)

    DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

```

FILLWITHCOMPUTERPLAYERS_FIELD_NUMBER = 3

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

STARTEVENTTYPE_FIELD_NUMBER = 2

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

StartEventType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

fillWithComputerPlayers

Magic attribute generated for “fillWithComputerPlayers” proto field.

gameId

Magic attribute generated for “gameId” proto field.

rejoinEvent = 1

startEvent = 0

startEventType

Magic attribute generated for “startEventType” proto field.

class pokerthproto.pokerth_pb2.**StartKickPetitionMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)
Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

KICKPLAYERID_FIELD_NUMBER = 4

KICKTIMEOUTSEC_FIELD_NUMBER = 5

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

NUMVOTESNEEDEDTOKICK_FIELD_NUMBER = 6

PETITIONID_FIELD_NUMBER = 2

PROPOSINGPLAYERID_FIELD_NUMBER = 3

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

gameId
Magic attribute generated for “gameId” proto field.

kickPlayerId
Magic attribute generated for “kickPlayerId” proto field.

kickTimeoutSec
Magic attribute generated for “kickTimeoutSec” proto field.

numVotesNeededToKick
Magic attribute generated for “numVotesNeededToKick” proto field.

petitionId
Magic attribute generated for “petitionId” proto field.

proposingPlayerId

Magic attribute generated for “proposingPlayerId” proto field.

class pokerthproto.pokerth_pb2.**StatisticsMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

STATISTICSDATA_FIELD_NUMBER = 1

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

class StatisticsData (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)
Checks if all required fields of a message are set.

Args:
errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

STATISTICSTYPE_FIELD_NUMBER = 1

STATISTICSVALUE_FIELD_NUMBER = 2

SerializePartialToString ()

SerializeToString ()

SetInParent ()
Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

StatisticsType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

statNumberOfPlayers = 1

statisticsType
Magic attribute generated for “statisticsType” proto field.

statisticsValue
Magic attribute generated for “statisticsValue” proto field.

StatisticsMessage.WhichOneof (*oneof_name*)
Returns the name of the currently set field inside a oneof, or None.

StatisticsMessage.statisticsData
Magic attribute generated for “statisticsData” proto field.

class pokerthproto.pokerth_pb2.**SubscriptionRequestMessage** (**kwargs)
Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()
Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

static RegisterExtension (*extension_handle*)

SUBSCRIPTIONACTION_FIELD_NUMBER = 1

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

SubscriptionAction = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

resubscribeGameList = 2

subscriptionAction

Magic attribute generated for “subscriptionAction” proto field.

unsubscribeGameList = 1

class pokerthproto.pokerth_pb2.**TimeoutWarningMessage** (***kwargs*)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REMAININGSECONDS_FIELD_NUMBER = 2

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

TIMEOUTREASON_FIELD_NUMBER = 1

TimeoutReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

remainingSeconds

Magic attribute generated for “remainingSeconds” proto field.

timeoutInactiveGame = 1

timeoutKickAfterAutofold = 2

timeoutNoDataReceived = 0

timeoutReason

Magic attribute generated for “timeoutReason” proto field.

class pokerthproto.pokerth_pb2.**UnknownAvatarMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

REQUESTID_FIELD_NUMBER = 1

static RegisterExtension (*extension_handle*)

SerializePartialToString ()

SerializeToString ()

SetInParent ()

Sets the `_cached_byte_size_dirty` bit to true, and propagates this to our listener iff this was a state change.

WhichOneof (*oneof_name*)

Returns the name of the currently set field inside a oneof, or None.

requestId

Magic attribute generated for “requestId” proto field.

class pokerthproto.pokerth_pb2.**VoteKickReplyMessage** (**kwargs)

Bases: google.protobuf.message.Message

ByteSize ()

Clear ()

ClearField (*field_name*)

DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>

FindInitializationErrors ()

Finds required fields which are not initialized.

Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g. “foo.bar[5].baz”.

static FromString (*s*)

GAMEID_FIELD_NUMBER = 1

HasField (*field_name*)

IsInitialized (*errors=None*)

Checks if all required fields of a message are set.

Args:

errors: A list which, if provided, will be populated with the field paths of all missing required fields.

Returns: True iff the specified message has all required fields set.

ListFields ()

MergeFrom (*msg*)

MergeFromString (*serialized*)

PETITIONID_FIELD_NUMBER = 2

```

static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
VOTEKICKREPLYTYPE_FIELD_NUMBER = 3
VoteKickReplyType = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.

gameId
    Magic attribute generated for “gameId” proto field.

petitionId
    Magic attribute generated for “petitionId” proto field.

voteKickAck = 0
voteKickDeniedAlreadyVoted = 2
voteKickDeniedInvalid = 1
voteKickReplyType
    Magic attribute generated for “voteKickReplyType” proto field.
class pokerthproto.pokerth_pb2.VoteKickRequestMessage (**kwargs)
    Bases: google.protobuf.message.Message

ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.

    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.

static FromString (s)
GAMEID_FIELD_NUMBER = 1
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.

    Args:

        errors: A list which, if provided, will be populated with the field paths of all missing required
            fields.

    Returns: True iff the specified message has all required fields set.

ListFields ()
MergeFrom (msg)

```

```
MergeFromString (serialized)
PETITIONID_FIELD_NUMBER = 2
static RegisterExtension (extension_handle)
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
VOTEKICK_FIELD_NUMBER = 3
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
gameId
    Magic attribute generated for “gameId” proto field.
petitionId
    Magic attribute generated for “petitionId” proto field.
voteKick
    Magic attribute generated for “voteKick” proto field.
class pokerthproto.pokerth_pb2.YourActionRejectedMessage (**kwargs)
    Bases: google.protobuf.message.Message
ByteSize ()
Clear ()
ClearField (field_name)
DESCRIPTOR = <google.protobuf.descriptor.Descriptor object>
FindInitializationErrors ()
    Finds required fields which are not initialized.
    Returns: A list of strings. Each string is a path to an uninitialized field from the top-level message, e.g.
        “foo.bar[5].baz”.
static FromString (s)
GAMEID_FIELD_NUMBER = 1
GAMESTATE_FIELD_NUMBER = 2
HasField (field_name)
IsInitialized (errors=None)
    Checks if all required fields of a message are set.
Args:
    errors: A list which, if provided, will be populated with the field paths of all missing required
        fields.
Returns: True iff the specified message has all required fields set.
ListFields ()
MergeFrom (msg)
MergeFromString (serialized)
```

```

REJECTIONREASON_FIELD_NUMBER = 5
static RegisterExtension (extension_handle)
RejectionReason = <google.protobuf.internal.enum_type_wrapper.EnumTypeWrapper object>
SerializePartialToString ()
SerializeToString ()
SetInParent ()
    Sets the _cached_byte_size_dirty bit to true, and propagates this to our listener iff this was a state change.
WhichOneof (oneof_name)
    Returns the name of the currently set field inside a oneof, or None.
YOURACTION_FIELD_NUMBER = 3
YOURRELATIVEBET_FIELD_NUMBER = 4
gameId
    Magic attribute generated for “gameId” proto field.
gameState
    Magic attribute generated for “gameState” proto field.
rejectedActionNotAllowed = 3
rejectedInvalidGameState = 1
rejectedNotYourTurn = 2
rejectionReason
    Magic attribute generated for “rejectionReason” proto field.
yourAction
    Magic attribute generated for “yourAction” proto field.
yourRelativeBet
    Magic attribute generated for “yourRelativeBet” proto field.

```

pokerthproto.protocol module

The PokerTH protocol consisting of messages and replies with respect to the current state in the communication.

```

class pokerthproto.protocol.ClientProtocol
    Bases: pokerthproto.protocol.PokerTHProtocol
afterHandShowCardsReceived (msg)
allInShowCardsReceived (msg)
announceReceived (msg)
chatReceived (msg)
dealFlopCardsReceived (msg)
dealRiverCardReceived (msg)
dealTurnCardReceived (msg)
endOfGameReceived (msg)
endOfHandHideCardsReceived (msg)

```

endOfHandShowCardsReceived (*msg*)

gameListNewReceived (*msg*)

gameListPlayerJoinedReceived (*msg*)

gameListPlayerLeftReceived (*msg*)

gameListUpdateReceived (*msg*)

gamePlayerJoinedReceived (*msg*)

gamePlayerLeftReceived (*msg*)

gameStartInitialReceived (*msg*)

handStartReceived (*msg*)

handleChat (*chatType*, *text*, *lobbyInfo*, *gameInfo=None*, *playerInfo=None*)

Handle the behavior of our client when a chat message was received.

Overwrite this method.

Parameters

- **chatType** – “Lobby”, “Game”, “Bot”, “Broadcast” or “Private”
- **text** – text of the message
- **lobbyInfo** – lobby information (*Lobby*)
- **gameInfo** – optional game information (*Game*)
- **playerInfo** – optional player information (*Player*)

handleEndOfGame (*gameInfo*, *winner*)

Handle the end of a game

The end of a game brings you back to the lobby

Parameters

- **gameInfo** – game information (*Game*)
- **winner** – winner of the game (*Player*)

handleEndOfHand (*gameInfo*)

Handle the end of a hand

Parameters **gameInfo** – game information (*Game*)

handleInsideLobby (*lobbyInfo*)

Handle the behavior of our client in the lobby.

Overwrite this method.

Parameters **lobbyInfo** – information about the lobby (*Lobby*)

handleMyTurn (*gameInfo*)

Decide what action to take when it is our turn.

Parameters **gameInfo** – game information (*Game*)

handleOthersTurn (*playerInfo*, *gameInfo*)

Handle the start of another player’s turn.

Parameters

- **playerInfo** – player information (*Player*)

- **gameInfo** – game information (*Game*)

initAckReceived (*msg*)

joinGameAckReceived (*msg*)

playerInfoReplyReceived (*msg*)

playerListReceived (*msg*)

playersActionDoneReceived (*msg*)

playersTurnReceived (*msg*)

sendChatRequest (*text*, *gameId=None*, *playerId=None*)

Send a chat message.

Parameters

- **text** – your message
- **gameId** – optional game id
- **playerId** – optional player id

sendJoinExistingGame (*gameId*, *autoLeave=False*)

sendJoinNewGame (*gameInfo*, *password=None*, *autoLeave=False*)

sendMyAction (*action*, *bet*, *relative=True*)

Send my action during a poker game.

Parameters

- **action** – action of *Action*
- **bet** – bet with respect to the action
- **relative** – boolean if the bet is relative to the highest set bet

sendStartEvent (*gameId*, *startEventType=None*, *fillWithBots=False*)

showMyCardsRequestReceived (*msg*)

startEventReceived (*msg*)

state = 0

yourActionRejected (*msg*)

class pokerthproto.protocol.**ClientProtocolFactory** (*nickName*)

Bases: twisted.internet.protocol.ClientFactory

clientConnectionFailed (*connector*, *reason*)

clientConnectionLost (*connector*, *reason*)

protocol

alias of *ClientProtocol*

class pokerthproto.protocol.**PokerTHProtocol**

Bases: twisted.internet.protocol.Protocol

adminBanPlayerAckReceived (*msg*)

adminBanPlayerReceived (*msg*)

adminRemoveGameAckReceived (*msg*)

adminRemoveGameReceived (*msg*)

afterHandShowCardsReceived (*msg*)
allInShowCardsReceived (*msg*)
announceReceived (*msg*)
askKickDeniedReceived (*msg*)
askKickPlayerReceived (*msg*)
authClientResponseReceived (*msg*)
authServerChallengeReceived (*msg*)
authServerVerificationReceived (*msg*)
avatarDataReceived (*msg*)
avatarEndReceived (*msg*)
avatarHeaderReceived (*msg*)
avatarRequestReceived (*msg*)
chatReceived (*msg*)
chatRejectReceived (*msg*)
chatRequestReceived (*msg*)
connectionLost (*reason*)
connectionMade ()
dataReceived (*data*)
dealFlopCardsReceived (*msg*)
dealRiverCardReceived (*msg*)
dealTurnCardReceived (*msg*)
dialogReceived (*msg*)
endKickPetitionReceived (*msg*)
endOfGameReceived (*msg*)
endOfHandHideCardsReceived (*msg*)
endOfHandShowCardsReceived (*msg*)
errorReceived (*msg*)
gameAdminChangedReceived (*msg*)
gameListAdminChangedReceived (*msg*)
gameListNewReceived (*msg*)
gameListPlayerJoinedReceived (*msg*)
gameListPlayerLeftReceived (*msg*)
gameListSpectatorJoinedReceived (*msg*)
gameListSpectatorLeftReceived (*msg*)
gameListUpdateReceived (*msg*)
gamePlayerJoinedReceived (*msg*)

gamePlayerLeftReceived (*msg*)
gameSpectatorJoinedReceived (*msg*)
gameSpectatorLeftReceived (*msg*)
gameStartInitialReceived (*msg*)
gameStartRejoinReceived (*msg*)
handStartReceived (*msg*)
initAckReceived (*msg*)
initReceived (*msg*)
inviteNotifyReceived (*msg*)
invitePlayerToGameReceived (*msg*)
joinExistingGameReceived (*msg*)
joinGameAckReceived (*msg*)
joinGameFailedReceived (*msg*)
joinNewGameReceived (*msg*)
kickPetitionUpdateReceived (*msg*)
kickPlayerRequestReceived (*msg*)
leaveGameRequestReceived (*msg*)
myActionRequestReceived (*msg*)
playerIdChangedReceived (*msg*)
playerInfoReplyReceived (*msg*)
playerInfoRequestReceived (*msg*)
playerListReceived (*msg*)
playersActionDoneReceived (*msg*)
playersTurnReceived (*msg*)
rejectGameInvitationReceived (*msg*)
rejectInvNotifyReceived (*msg*)
rejoinExistingGameReceived (*msg*)
removedFromGameReceived (*msg*)
reportAvatarAckReceived (*msg*)
reportAvatarReceived (*msg*)
reportGameAckReceived (*msg*)
reportGameReceived (*msg*)
resetTimeoutReceived (*msg*)
showMyCardsRequestReceived (*msg*)
startEventAckReceived (*msg*)
startEventReceived (*msg*)

```
startKickPetitionReceived (msg)
statisticsReceived (msg)
subscriptionRequestReceived (msg)
timeoutWarningReceived (msg)
unhandledMessageReceived (msg)
unknownAvatarReceived (msg)
voteKickReplyReceived (msg)
voteKickRequestReceived (msg)
yourActionRejectedReceived (msg)
```

```
class pokerthproto.protocol.States
```

```
  Bases: object
```

```
  Enum of all client states
```

```
  GAME_JOINED = 2
```

```
  GAME_STARTED = 3
```

```
  INIT = 0
```

```
  LOBBY = 1
```

```
pokerthproto.protocol.enum2str (enumType, enum)
```

```
  Translates a pokerth_pb2 enum type to a string.
```

Parameters

- **enumType** – enum type class
- **enum** – the enum element of the type

Returns identifier string of enum

pokerthproto.proxy module

A PokerTH proxy that logs all messages between a PokerTH client and server.

```
class pokerthproto.proxy.ClientProtocol
```

```
  Bases: pokerthproto.protocol.PokerTHProtocol
```

```
  dataReceived (data)
```

```
class pokerthproto.proxy.ClientProtocolFactory (sendToClient)
```

```
  Bases: twisted.internet.protocol.Factory
```

```
  protocol
```

```
    alias of ClientProtocol
```

```
class pokerthproto.proxy.ProxyProtocol
```

```
  Bases: pokerthproto.protocol.PokerTHProtocol
```

```
  connectionMade ()
```

```
  dataReceived (data)
```

```
  registerServer (proto)
```

```
  sendToClient (data)
```

```
class pokerthproto.proxy.ProxyProtocolFactory
    Bases: twisted.internet.protocol.Factory

    protocol
        alias of ProxyProtocol
```

pokerthproto.transport module

The low-level transport functionality for packing/unpacking and enveloping/ developing messages.

```
pokerthproto.transport.develop(envelope)
```

Remove the envelope from a message.

Parameters `envelope` – PokerTHMessage object that envelops a message

Returns PokerTH message from the envelope

```
pokerthproto.transport.envelop(msg)
```

Put a message into an envelope.

Parameters `msg` – PokerTH message object

Returns message wrapped in a PokerTHMessage object

```
pokerthproto.transport.makeSizeBytes(n)
```

Create a 4 bytes string that encodes the number n.

Parameters `n` – integer

Returns 4 bytes string

```
pokerthproto.transport.pack(envelope)
```

Packs/Serializes a PokerTHMessage to a data string.

Parameters `envelope` – PokerTHMessage envelope

Returns data as string

```
pokerthproto.transport.readSizeBytes(string)
```

Reads the 4 byte size-string and returns the size as integer.

Parameters `string` – 4 byte size-string

Returns integer

```
pokerthproto.transport.unpack(data)
```

Unpacks/Deserializes a PokerTH network message.

Parameters `data` – data as string

Returns PokerTHMessage object containing the message

Module contents

Indices and tables

- `genindex`
- `modindex`
- `search`

p

`pokerthproto`, 113
`pokerthproto.game`, 7
`pokerthproto.lobby`, 9
`pokerthproto.player`, 10
`pokerthproto.poker`, 11
`pokerthproto.pokerth_pb2`, 12
`pokerthproto.protocol`, 107
`pokerthproto.proxy`, 112
`pokerthproto.transport`, 113

A

- Action (class in pokerthproto.poker), 11
- ActionInfo (class in pokerthproto.game), 7
- actions (pokerthproto.game.RoundInfo attribute), 9
- addAction() (pokerthproto.game.Game method), 7
- addGameInfo() (pokerthproto.lobby.Lobby method), 10
- addOthersCards() (pokerthproto.game.Game method), 7
- addPlayer() (pokerthproto.game.Game method), 7
- addPlayer() (pokerthproto.lobby.GameInfo method), 9
- addPlayer() (pokerthproto.lobby.Lobby method), 10
- addPlayerToGame() (pokerthproto.lobby.Lobby method), 10
- addRound() (pokerthproto.game.Game method), 8
- addWin() (pokerthproto.game.Game method), 8
- AdminBanPlayerAckMessage (class in pokerthproto.pokerth_pb2), 12
- adminBanPlayerAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
- ADMINBANPLAYERACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
- adminBanPlayerAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 109
- AdminBanPlayerMessage (class in pokerthproto.pokerth_pb2), 13
- adminBanPlayerMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
- ADMINBANPLAYERMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
- adminBanPlayerReceived() (pokerthproto.protocol.PokerTHProtocol method), 109
- AdminBanPlayerResult (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage attribute), 12
- adminPlayerId (pokerthproto.lobby.GameInfo attribute), 9
- adminPlayerId (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42
- ADMINPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 41
- AdminRemoveGameAckMessage (class in pokerthproto.pokerth_pb2), 13
- adminRemoveGameAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
- ADMINREMOVEGAMEACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
- adminRemoveGameAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 109
- AdminRemoveGameMessage (class in pokerthproto.pokerth_pb2), 14
- adminRemoveGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
- ADMINREMOVEGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
- adminRemoveGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 109
- AdminRemoveGameResult (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 13
- AfterHandShowCardsMessage (class in pokerthproto.pokerth_pb2), 15
- afterHandShowCardsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
- AFTERHANDSHOWCARDSMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

afterHandShowCardsReceived() (pokerthproto.protocol.ClientProtocol method), 107
 afterHandShowCardsReceived() (pokerthproto.protocol.PokerTHProtocol method), 109
 ALLIN (pokerthproto.poker.Action attribute), 11
 allInCard1 (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 17
 ALLINCARD1_FIELD_NUMBER (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn attribute), 17
 allInCard2 (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 17
 ALLINCARD2_FIELD_NUMBER (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn attribute), 17
 AllInShowCardsMessage (class in pokerthproto.pokerth_pb2), 16
 allInShowCardsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 AllInShowCardsMessage.PlayerAllIn (class in pokerthproto.pokerth_pb2), 17
 ALLINSHOWCARDSMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 allInShowCardsReceived() (pokerthproto.protocol.ClientProtocol method), 107
 allInShowCardsReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 allowSpectators (pokerthproto.lobby.GameInfo attribute), 9
 allowSpectators (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
 ALLOWSPECTATORS_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 67
 AnnounceMessage (class in pokerthproto.pokerth_pb2), 18
 announceMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 AnnounceMessage.Version (class in pokerthproto.pokerth_pb2), 19
 ANNOUNCEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 announceReceived() (pokerthproto.protocol.ClientProtocol method), 107
 announceReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 areYouGameAdmin (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61
 AREYOUGAMEADMIN_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 60
 AskPlayDeniedMessage (class in pokerthproto.pokerth_pb2), 20
 askKickDeniedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 ASKKICKDENIEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 askKickDeniedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 AskKickPlayerMessage (class in pokerthproto.pokerth_pb2), 21
 askKickPlayerMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 ASKKICKPLAYERMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 askKickPlayerReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 AuthClientResponseMessage (class in pokerthproto.pokerth_pb2), 22
 authClientResponseMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 AUTHCLIENTRESPONSEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 authClientResponseReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 authenticatedLogin (pokerthproto.pokerth_pb2.InitMessage attribute), 57
 AuthServerChallengeMessage (class in pokerthproto.pokerth_pb2), 23
 authServerChallengeMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
 AUTHSERVERCHALLENGEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
 authServerChallengeReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

authServerPassword	(pokerthproto.pokerth_pb2.InitMessage attribute), 57	AvatarEndMessage (class in pokerthproto.pokerth_pb2), 25
AUTHSERVERPASSWORD_FIELD_NUMBER	(pokerthproto.pokerth_pb2.InitMessage attribute), 56	avatarEndMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
AuthServerVerificationMessage	(class in pokerthproto.pokerth_pb2), 23	AVATARENDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
authServerVerificationMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	avatarEndReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
AUTHSERVERVERIFICATIONMESSAGE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79	AvatarHash (pokerthproto.player.Player attribute), 10
authServerVerificationReceived()	(pokerthproto.protocol.PokerTHProtocol method), 110	avatarHash (pokerthproto.pokerth_pb2.AvatarRequestMessage attribute), 27
autoLeave (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60		avatarHash (pokerthproto.pokerth_pb2.InitMessage attribute), 57
autoLeave (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 63		avatarHash (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 72
autoLeave (pokerthproto.pokerth_pb2.RejoinExistingGameMessage attribute), 90		AVATARHASH_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarRequestMessage attribute), 27
AUTOLEAVE_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 59		AVATARHASH_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute), 56
AUTOLEAVE_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 63		AVATARHASH_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 71
AUTOLEAVE_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejoinExistingGameMessage attribute), 90		AvatarHeaderMessage (class in pokerthproto.pokerth_pb2), 26
avatarBlock (pokerthproto.pokerth_pb2.AvatarDataMessage attribute), 25		avatarHeaderMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
AVATARBLOCK_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarDataMessage attribute), 24		AVATARHEADERMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79
avatarData (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 73		avatarHeaderReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
AVATARDATA_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 71		avatarReportAccepted (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92
AvatarDataMessage (class in pokerthproto.pokerth_pb2), 24		avatarReportDuplicate (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92
avatarDataMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84		avatarReportInvalid (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92
AVATARDATAMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79		AvatarRequestMessage (class in pokerthproto.pokerth_pb2), 27
avatarDataReceived() (pokerthproto.protocol.PokerTHProtocol method), 110		avatarRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
		AVATARREQUESTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84

ByteSize() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 24	ByteSize() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 49
ByteSize() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25	ByteSize() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 50
ByteSize() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26	ByteSize() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 51
ByteSize() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27	ByteSize() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayer method), 52
ByteSize() (pokerthproto.pokerth_pb2.ChatMessage method), 28	ByteSize() (pokerthproto.pokerth_pb2.HandStartMessage method), 53
ByteSize() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29	ByteSize() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54
ByteSize() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30	ByteSize() (pokerthproto.pokerth_pb2.InitAckMessage method), 55
ByteSize() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31	ByteSize() (pokerthproto.pokerth_pb2.InitMessage method), 56
ByteSize() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32	ByteSize() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58
ByteSize() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 32	ByteSize() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59
ByteSize() (pokerthproto.pokerth_pb2.DialogMessage method), 33	ByteSize() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60
ByteSize() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34	ByteSize() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61
ByteSize() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 35	ByteSize() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 61
ByteSize() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 36	ByteSize() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63
ByteSize() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 37	ByteSize() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64
ByteSize() (pokerthproto.pokerth_pb2.ErrorMessage method), 38	ByteSize() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65
ByteSize() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 39	ByteSize() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
ByteSize() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 40	ByteSize() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 66
ByteSize() (pokerthproto.pokerth_pb2.GameListNewMessage method), 41	ByteSize() (pokerthproto.pokerth_pb2.NetGameInfo method), 68
ByteSize() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 42	ByteSize() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70
ByteSize() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 43	ByteSize() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71
ByteSize() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44	ByteSize() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoReply method), 72
ByteSize() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	ByteSize() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoReply method), 71
ByteSize() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46	ByteSize() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 73
ByteSize() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47	ByteSize() (pokerthproto.pokerth_pb2.PlayerListMessage method), 74
ByteSize() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48	ByteSize() (pokerthproto.pokerth_pb2.PlayerResult method), 75
ByteSize() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	ByteSize() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 76

ByteSize() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 75
 ByteSize() (pokerthproto.pokerth_pb2.PokerTHMessage method), 79
 ByteSize() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 85
 ByteSize() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88
 ByteSize() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89
 ByteSize() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90
 ByteSize() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 90
 ByteSize() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92
 ByteSize() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93
 ByteSize() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 93
 ByteSize() (pokerthproto.pokerth_pb2.ReportGameMessage method), 94
 ByteSize() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 95
 ByteSize() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96
 ByteSize() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97
 ByteSize() (pokerthproto.pokerth_pb2.StartEventMessage method), 97
 ByteSize() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 98
 ByteSize() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100
 ByteSize() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 100
 ByteSize() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 101
 ByteSize() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 102
 ByteSize() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 103
 ByteSize() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104
 ByteSize() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105
 ByteSize() (pokerthproto.pokerth_pb2.YourActionRejectedMessage method), 106

C

CALL (pokerthproto.poker.Action attribute), 11
 cards (pokerthproto.game.RoundInfo attribute), 9
 cardsValue (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
 CARDSVALUE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
 cardToInt() (in module pokerthproto.poker), 11
 ChatMessage (class in pokerthproto.pokerth_pb2), 28
 chatMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 28
 CHATMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 28
 ChatRejectReceived() (pokerthproto.protocol.ClientProtocol method), 107
 ChatRejectReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 ChatRejectMessage (class in pokerthproto.pokerth_pb2), 29
 chatRejectMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 29
 CHATREJECTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 29
 chatRejectReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 ChatRequestMessage (class in pokerthproto.pokerth_pb2), 30
 chatRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 30
 CHATREQUESTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 30
 chatRequestReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
 chatType (pokerthproto.pokerth_pb2.ChatMessage attribute), 28
 ChatType (pokerthproto.pokerth_pb2.ChatMessage attribute), 28
 chatType (pokerthproto.pokerth_pb2.ChatMessage attribute), 28
 CHATTEXT_FIELD_NUMBER (pokerthproto.pokerth_pb2.ChatMessage attribute), 28
 CHATTEXT_FIELD_NUMBER (pokerthproto.pokerth_pb2.ChatRequestMessage attribute), 30

CHATTYPE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.ChatMessage attribute), 28	Clear()	(pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
chatTypeBot	(pokerthproto.pokerth_pb2.ChatMessage attribute), 29	Clear()	(pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
chatTypeBroadcast	(pokerthproto.pokerth_pb2.ChatMessage attribute), 29	Clear()	(pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31
chatTypeGame	(pokerthproto.pokerth_pb2.ChatMessage attribute), 29	Clear()	(pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32
chatTypeLobby	(pokerthproto.pokerth_pb2.ChatMessage attribute), 29	Clear()	(pokerthproto.pokerth_pb2.DealTurnCardMessage method), 32
chatTypePrivate	(pokerthproto.pokerth_pb2.ChatMessage attribute), 29	Clear()	(pokerthproto.pokerth_pb2.DialogMessage method), 33
CHECK	(pokerthproto.poker.Action attribute), 11	Clear()	(pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34
Clear()	(pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12	Clear()	(pokerthproto.pokerth_pb2.EndOfGameMessage method), 35
Clear()	(pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13	Clear()	(pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 36
Clear()	(pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14	Clear()	(pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 37
Clear()	(pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 14	Clear()	(pokerthproto.pokerth_pb2.ErrorMessage method), 38
Clear()	(pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 15	Clear()	(pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 39
Clear()	(pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16	Clear()	(pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 40
Clear()	(pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 17	Clear()	(pokerthproto.pokerth_pb2.GameListNewMessage method), 41
Clear()	(pokerthproto.pokerth_pb2.AnnounceMessage method), 18	Clear()	(pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 42
Clear()	(pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19	Clear()	(pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 43
Clear()	(pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20	Clear()	(pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44
Clear()	(pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21	Clear()	(pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45
Clear()	(pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22	Clear()	(pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46
Clear()	(pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23	Clear()	(pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47
Clear()	(pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 23	Clear()	(pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48
Clear()	(pokerthproto.pokerth_pb2.AvatarDataMessage method), 24	Clear()	(pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49
Clear()	(pokerthproto.pokerth_pb2.AvatarEndMessage method), 25	Clear()	(pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50
Clear()	(pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26	Clear()	(pokerthproto.pokerth_pb2.GameStartInitialMessage method), 50
Clear()	(pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27	Clear()	(pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 51
Clear()	(pokerthproto.pokerth_pb2.ChatMessage method), 28	Clear()	(pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayer method), 52
		Clear()	(pokerthproto.pokerth_pb2.HandStartMessage method), 53

Clear() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54

Clear() (pokerthproto.pokerth_pb2.InitAckMessage method), 55

Clear() (pokerthproto.pokerth_pb2.InitMessage method), 56

Clear() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58

Clear() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59

Clear() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60

Clear() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61

Clear() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62

Clear() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63

Clear() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64

Clear() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65

Clear() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66

Clear() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 66

Clear() (pokerthproto.pokerth_pb2.NetGameInfo method), 68

Clear() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70

Clear() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71

Clear() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 72

Clear() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoDataAvatarData method), 71

Clear() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 73

Clear() (pokerthproto.pokerth_pb2.PlayerListMessage method), 74

Clear() (pokerthproto.pokerth_pb2.PlayerResult method), 75

Clear() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 76

Clear() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78

Clear() (pokerthproto.pokerth_pb2.PokerTHMessage method), 79

Clear() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88

Clear() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89

Clear() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90

Clear() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91

Clear() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92

Clear() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93

Clear() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 93

Clear() (pokerthproto.pokerth_pb2.ReportGameMessage method), 94

Clear() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 95

Clear() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96

Clear() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97

Clear() (pokerthproto.pokerth_pb2.StartEventMessage method), 97

Clear() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 98

Clear() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100

Clear() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 100

Clear() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 101

Clear() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 102

Clear() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 103

Clear() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104

Clear() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105

Clear() (pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 106

ClearField() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12

ClearField() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13

ClearField() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14

ClearField() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 14

ClearField() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 15

ClearField() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16

ClearField() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17

ClearField() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18

ClearField() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19

ClearField() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20

ClearField() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21

ClearField() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22

ClearField() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23

ClearField() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 23

ClearField() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 24

ClearField() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25

ClearField() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26

ClearField() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27

ClearField() (pokerthproto.pokerth_pb2.ChatMessage method), 28

ClearField() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29

ClearField() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30

ClearField() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31

ClearField() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32

ClearField() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 32

ClearField() (pokerthproto.pokerth_pb2.DialogMessage method), 33

ClearField() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34

ClearField() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 35

ClearField() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 36

ClearField() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 37

ClearField() (pokerthproto.pokerth_pb2.ErrorMessage method), 38

ClearField() (pokerthproto.pokerth_pb2.GameAdminChangeMessage method), 39

ClearField() (pokerthproto.pokerth_pb2.GameListAdminChangeMessage method), 40

ClearField() (pokerthproto.pokerth_pb2.GameListNewMessage method), 41

ClearField() (pokerthproto.pokerth_pb2.GameListPlayerJoinMessage method), 42

ClearField() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 43

ClearField() (pokerthproto.pokerth_pb2.GameListSpectatorJoinMessage method), 44

ClearField() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45

ClearField() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46

ClearField() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47

ClearField() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48

ClearField() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49

ClearField() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50

ClearField() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 50

ClearField() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 51

ClearField() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayer method), 52

ClearField() (pokerthproto.pokerth_pb2.HandStartMessage method), 53

ClearField() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54

ClearField() (pokerthproto.pokerth_pb2.InitAckMessage method), 55

ClearField() (pokerthproto.pokerth_pb2.InitMessage method), 56

ClearField() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58

ClearField() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59

ClearField() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60

ClearField() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61

ClearField() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62

ClearField() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63

ClearField() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64

ClearField() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65

ClearField() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66

ClearField() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 66

ClearField() (pokerthproto.pokerth_pb2.NetGameInfo method), 68

ClearField() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70

ClearField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71

ClearField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfo method), 72

ClearField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 71	ClearField() (pokerthproto.pokerth_pb2.YourActionRejectedMessage attribute), 106
ClearField() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage attribute), 73	ClientConnectionFailed() (pokerthproto.protocol.ClientProtocolFactory method), 109
ClearField() (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 74	clientConnectionLost() (pokerthproto.protocol.ClientProtocolFactory method), 109
ClearField() (pokerthproto.pokerth_pb2.PlayerResult attribute), 75	ClientProtocol (class in pokerthproto.protocol), 107
ClearField() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 76	ClientProtocol (class in pokerthproto.proxy), 112
ClearField() (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78	ClientProtocolFactory (class in pokerthproto.protocol), 109
ClearField() (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79	ClientProtocolFactory (class in pokerthproto.proxy), 112
ClearField() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 88	clientResponse (pokerthproto.pokerth_pb2.AuthClientResponseMessage attribute), 22
ClearField() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 89	CLIENTRESPONSE_FIELD_NUMBER (pokerthproto.pokerth_pb2.AuthClientResponseMessage attribute), 22
ClearField() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage attribute), 90	clientUserData (pokerthproto.pokerth_pb2.InitMessage attribute), 57
ClearField() (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91	CLIENTUSERDATA_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute), 56
ClearField() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92	connectionLost() (pokerthproto.protocol.PokerTHProtocol method), 110
ClearField() (pokerthproto.pokerth_pb2.ReportAvatarMessage attribute), 93	connectionMade() (pokerthproto.proxy.ProxyProtocol method), 112
ClearField() (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 93	countryCode (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 73
ClearField() (pokerthproto.pokerth_pb2.ReportGameMessage attribute), 94	COUNTRYCODE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 72
ClearField() (pokerthproto.pokerth_pb2.ResetTimeoutMessage attribute), 95	currRound (pokerthproto.game.Game attribute), 8
ClearField() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage attribute), 96	currRoundInfo (pokerthproto.game.Game attribute), 8
ClearField() (pokerthproto.pokerth_pb2.StartEventAckMessage attribute), 97	D
ClearField() (pokerthproto.pokerth_pb2.StartEventMessage attribute), 97	dealerReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
ClearField() (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 98	dealer() (pokerthproto.proxy.ClientProtocol method), 112
ClearField() (pokerthproto.pokerth_pb2.StatisticsMessage attribute), 100	dealerReceived() (pokerthproto.proxy.ProxyProtocol method), 112
ClearField() (pokerthproto.pokerth_pb2.StatisticsMessage attribute), 100	dealer (pokerthproto.game.Game attribute), 8
ClearField() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage attribute), 101	dealerPlayerId (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
ClearField() (pokerthproto.pokerth_pb2.TimeoutWarningMessage attribute), 102	DEALERPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
ClearField() (pokerthproto.pokerth_pb2.UnknownAvatarMessage attribute), 103	
ClearField() (pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 104	
ClearField() (pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 105	

tribute), 53	DESCRIPTOR (pokerth-
DealFlopCardsMessage (class in pokerth-	proto.pokerth_pb2.AdminBanPlayerAckMessage
proto.pokerth_pb2), 31	attribute), 12
dealFlopCardsMessage (pokerth-	DESCRIPTOR (pokerth-
proto.pokerth_pb2.PokerTHMessage attribute),	proto.pokerth_pb2.AdminBanPlayerMessage
85	attribute), 13
DEALFLOPCARDSMESSAGE_FIELD_NUMBER	DESCRIPTOR (pokerth-
(pokerthproto.pokerth_pb2.PokerTHMessage	proto.pokerth_pb2.AdminRemoveGameAckMessage
attribute), 79	attribute), 14
dealFlopCardsReceived() (pokerth-	DESCRIPTOR (pokerth-
proto.protocol.ClientProtocol method), 107	proto.pokerth_pb2.AdminRemoveGameMessage
dealFlopCardsReceived() (pokerth-	attribute), 14
proto.protocol.PokerTHProtocol method),	DESCRIPTOR (pokerth-
110	proto.pokerth_pb2.AfterHandShowCardsMessage
DealRiverCardMessage (class in pokerth-	attribute), 15
proto.pokerth_pb2), 32	DESCRIPTOR (pokerth-
dealRiverCardMessage (pokerth-	proto.pokerth_pb2.AllInShowCardsMessage
proto.pokerth_pb2.PokerTHMessage attribute),	attribute), 16
85	DESCRIPTOR (pokerth-
DEALRIVERCARDMESSAGE_FIELD_NUMBER	proto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn
(pokerthproto.pokerth_pb2.PokerTHMessage	attribute), 17
attribute), 79	DESCRIPTOR (pokerth-
dealRiverCardReceived() (pokerth-	proto.pokerth_pb2.AnnounceMessage at-
proto.protocol.ClientProtocol method), 107	tribute), 18
dealRiverCardReceived() (pokerth-	DESCRIPTOR (pokerth-
proto.protocol.PokerTHProtocol method),	proto.pokerth_pb2.AnnounceMessage.Version
110	attribute), 19
DealTurnCardMessage (class in pokerth-	DESCRIPTOR (pokerth-
proto.pokerth_pb2), 32	proto.pokerth_pb2.AskKickDeniedMessage
dealTurnCardMessage (pokerth-	attribute), 20
proto.pokerth_pb2.PokerTHMessage attribute),	DESCRIPTOR (pokerth-
85	proto.pokerth_pb2.AskKickPlayerMessage
DEALTURNCARDMESSAGE_FIELD_NUMBER	attribute), 21
(pokerthproto.pokerth_pb2.PokerTHMessage	DESCRIPTOR (pokerth-
attribute), 79	proto.pokerth_pb2.AuthClientResponseMessage
dealTurnCardReceived() (pokerth-	attribute), 22
proto.protocol.ClientProtocol method), 107	DESCRIPTOR (pokerth-
dealTurnCardReceived() (pokerth-	proto.pokerth_pb2.AuthServerChallengeMessage
proto.protocol.PokerTHProtocol method),	attribute), 23
110	DESCRIPTOR (pokerth-
delayBetweenHands (pokerthproto.lobby.GameInfo at-	proto.pokerth_pb2.AuthServerVerificationMessage
tribute), 9	attribute), 23
delayBetweenHands (pokerth-	DESCRIPTOR (pokerth-
proto.pokerth_pb2.NetGameInfo attribute),	proto.pokerth_pb2.AvatarDataMessage at-
69	tribute), 24
DELAYBETWEENHANDS_FIELD_NUMBER (pok-	DESCRIPTOR (pokerth-
erthproto.pokerth_pb2.NetGameInfo attribute),	proto.pokerth_pb2.AvatarEndMessage at-
68	tribute), 25
delPlayer() (pokerthproto.game.Game method), 8	DESCRIPTOR (pokerth-
delPlayer() (pokerthproto.lobby.GameInfo method), 9	proto.pokerth_pb2.AvatarHeaderMessage
delPlayer() (pokerthproto.lobby.Lobby method), 10	attribute), 26
delPlayerFromGame() (pokerthproto.lobby.Lobby	DESCRIPTOR (pokerth-
method), 10	proto.pokerth_pb2.AvatarRequestMessage
	attribute), 27

DESCRIPTOR (pokerthproto.pokerth_pb2.ChatMessage attribute), 28	DESCRIPTOR (pokerthproto.pokerth_pb2.ChatMessage attribute), 45
DESCRIPTOR (pokerthproto.pokerth_pb2.ChatRejectMessage attribute), 29	DESCRIPTOR (pokerthproto.pokerth_pb2.GameListUpdateMessage attribute), 46
DESCRIPTOR (pokerthproto.pokerth_pb2.ChatRequestMessage attribute), 30	DESCRIPTOR (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47
DESCRIPTOR (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31	DESCRIPTOR (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48
DESCRIPTOR (pokerthproto.pokerth_pb2.DealRiverCardMessage attribute), 32	DESCRIPTOR (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage attribute), 49
DESCRIPTOR (pokerthproto.pokerth_pb2.DealTurnCardMessage attribute), 33	DESCRIPTOR (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50
DESCRIPTOR (pokerthproto.pokerth_pb2.DialogMessage attribute), 33	DESCRIPTOR (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51
DESCRIPTOR (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 34	DESCRIPTOR (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 51
DESCRIPTOR (pokerthproto.pokerth_pb2.EndOfGameMessage attribute), 35	DESCRIPTOR (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData attribute), 52
DESCRIPTOR (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 36	DESCRIPTOR (pokerthproto.pokerth_pb2.HandStartMessage attribute), 53
DESCRIPTOR (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage attribute), 37	DESCRIPTOR (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards attribute), 54
DESCRIPTOR (pokerthproto.pokerth_pb2.ErrorMessage attribute), 38	DESCRIPTOR (pokerthproto.pokerth_pb2.InitAckMessage attribute), 55
DESCRIPTOR (pokerthproto.pokerth_pb2.GameAdminChangedMessage attribute), 39	DESCRIPTOR (pokerthproto.pokerth_pb2.InitMessage attribute), 56
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListAdminChangedMessage attribute), 40	DESCRIPTOR (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 41	DESCRIPTOR (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage attribute), 59
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage attribute), 42	DESCRIPTOR (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage attribute), 43	DESCRIPTOR (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage attribute), 44	DESCRIPTOR (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62
DESCRIPTOR (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage attribute), 45	DESCRIPTOR (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 63

DESCRIPTOR	(pokerth- proto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ReportAvatarAckMessage attribute), 91
DESCRIPTOR	(pokerth- proto.pokerth_pb2.KickPlayerRequestMessage attribute), 65	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ReportAvatarMessage attribute), 92
DESCRIPTOR	(pokerth- proto.pokerth_pb2.LeaveGameRequestMessage attribute), 66	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ReportGameAckMessage attribute), 93
DESCRIPTOR	(pokerth- proto.pokerth_pb2.MyActionRequestMessage attribute), 66	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ReportGameMessage attribute), 93
DESCRIPTOR	(pokerthproto.pokerth_pb2.NetGameInfo attribute), 68	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ResetTimeoutMessage attribute), 94
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerIdChangedMessage attribute), 70	DESCRIPTOR	(pokerth- proto.pokerth_pb2.ShowMyCardsRequestMessage attribute), 95
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerInfoReplyMessage attribute), 71	DESCRIPTOR	(pokerth- proto.pokerth_pb2.StartEventAckMessage attribute), 96
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 72	DESCRIPTOR	(pokerth- proto.pokerth_pb2.StartEventMessage attribute), 97
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 71	DESCRIPTOR	(pokerth- proto.pokerth_pb2.StartKickPetitionMessage attribute), 98
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerInfoRequestMessage attribute), 73	DESCRIPTOR	(pokerth- proto.pokerth_pb2.StatisticsMessage attribute), 99
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayerListMessage attribute), 74	DESCRIPTOR	(pokerth- proto.pokerth_pb2.StatisticsMessage.StatisticsData attribute), 100
DESCRIPTOR	(pokerthproto.pokerth_pb2.PlayerResult attribute), 75	DESCRIPTOR	(pokerth- proto.pokerth_pb2.SubscriptionRequestMessage attribute), 101
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayersActionDoneMessage attribute), 76	DESCRIPTOR	(pokerth- proto.pokerth_pb2.TimeoutWarningMessage attribute), 102
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PlayersTurnMessage attribute), 78	DESCRIPTOR	(pokerth- proto.pokerth_pb2.UnknownAvatarMessage attribute), 103
DESCRIPTOR	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 79	DESCRIPTOR	(pokerth- proto.pokerth_pb2.VoteKickReplyMessage attribute), 104
DESCRIPTOR	(pokerth- proto.pokerth_pb2.RejectGameInvitationMessage attribute), 88	DESCRIPTOR	(pokerth- proto.pokerth_pb2.VoteKickRequestMessage attribute), 105
DESCRIPTOR	(pokerth- proto.pokerth_pb2.RejectInvNotifyMessage attribute), 89	DESCRIPTOR	(pokerth- proto.pokerth_pb2>YourActionRejectedMessage attribute), 106
DESCRIPTOR	(pokerth- proto.pokerth_pb2.RejoinExistingGameMessage attribute), 90	DESCRIPTOR	develop() (in module pokerthproto.transport), 113
DESCRIPTOR	(pokerth- proto.pokerth_pb2.RemovedFromGameMessage attribute), 91	DESCRIPTOR	DialogMessage (class in pokerthproto.pokerth_pb2), 33

dialogMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

DIALOGMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

dialogReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

doubleBlinds (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69

E

encryptedCards (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55

ENCRYPTEDCARDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 53

EndKickPetitionMessage (class in pokerthproto.pokerth_pb2), 34

endKickPetitionMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

ENDKICKPETITIONMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

endKickPetitionReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

EndOfGameMessage (class in pokerthproto.pokerth_pb2), 35

endOfGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

ENDOFGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

endOfGameReceived() (pokerthproto.protocol.ClientProtocol method), 107

endOfGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

EndOfHandHideCardsMessage (class in pokerthproto.pokerth_pb2), 36

endOfHandHideCardsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

ENDOFHANDHIDECARDSMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

endOfHandHideCardsReceived() (pokerthproto.protocol.ClientProtocol method), 107

endOfHandHideCardsReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

EndOfHandShowCardsMessage (class in pokerthproto.pokerth_pb2), 37

endOfHandShowCardsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

ENDOFHANDSHOWCARDSMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

endOfHandShowCardsReceived() (pokerthproto.protocol.ClientProtocol method), 107

endOfHandShowCardsReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

endRaiseMode (pokerthproto.lobby.GameInfo attribute), 9

EndRaiseMode (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68

endRaiseMode (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69

ENDRAISEMODE_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68

endRaiseSmallBlindValue (pokerthproto.lobby.GameInfo attribute), 9

endRaiseSmallBlindValue (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69

ENDRAISESMALLBLINDVALUE_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68

enum2str() (in module pokerthproto.protocol), 112

envelop() (in module pokerthproto.transport), 113

ErrorMessage (class in pokerthproto.pokerth_pb2), 38

errorMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

ERRORMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 79

ErrorReason (pokerthproto.pokerth_pb2.ErrorMessage attribute), 38

errorReason (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39

ERRORREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.ErrorMessage attribute), 38

errorReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

existPlayer() (pokerthproto.game.Game method), 8

existRound() (pokerthproto.game.Game method), 8

F

fillWithComputerPlayers (pokerthproto.lobby.GameInfo attribute), 9	FindInitializationErrors() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25
fillWithComputerPlayers (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98	FindInitializationErrors() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26
FILLWITHCOMPUTERPLAYERS_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartEventMessage attribute), 97	FindInitializationErrors() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27
FindInitializationErrors() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12	FindInitializationErrors() (pokerthproto.pokerth_pb2.ChatMessage method), 28
FindInitializationErrors() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13	FindInitializationErrors() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
FindInitializationErrors() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14	FindInitializationErrors() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
FindInitializationErrors() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15	FindInitializationErrors() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31
FindInitializationErrors() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 15	FindInitializationErrors() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32
FindInitializationErrors() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16	FindInitializationErrors() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33
FindInitializationErrors() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17	FindInitializationErrors() (pokerthproto.pokerth_pb2.DialogMessage method), 33
FindInitializationErrors() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18	FindInitializationErrors() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34
FindInitializationErrors() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19	FindInitializationErrors() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 35
FindInitializationErrors() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20	FindInitializationErrors() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 36
FindInitializationErrors() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21	FindInitializationErrors() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 37
FindInitializationErrors() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22	FindInitializationErrors() (pokerthproto.pokerth_pb2.ErrorMessage method), 38
FindInitializationErrors() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23	FindInitializationErrors() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 39
FindInitializationErrors() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 23	FindInitializationErrors() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 40
FindInitializationErrors() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 24	FindInitializationErrors() (pokerthproto.pokerth_pb2.GameListNewMessage method), 41

FindInitializationErrors() proto.pokerth_pb2.GameListPlayerJoinedMessage method), 42	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.JoinExistingGameMessage method), 60	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameListPlayerLeftMessage method), 43	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.JoinGameAckMessage method), 61	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.JoinGameFailedMessage method), 62	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.JoinNewGameMessage method), 63	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameListUpdateMessage method), 46	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.KickPetitionUpdateMessage method), 64	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GamePlayerJoinedMessage method), 47	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.KickPlayerRequestMessage method), 65	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GamePlayerLeftMessage method), 48	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.LeaveGameRequestMessage method), 66	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.MyActionRequestMessage method), 67	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameSpectatorLeftMessage method), 50	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.NetGameInfo method), 68	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameStartInitialMessage method), 51	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerIdChangedMessage method), 70	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameStartRejoinMessage method), 51	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerInfoReplyMessage method), 71	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 52	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 72	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.HandStartMessage method), 53	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.Avat method), 71	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.HandStartMessage.PlainCards method), 54	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerInfoRequestMessage method), 73	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.InitAckMessage method), 55	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerListMessage method), 74	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.InitMessage method), 56	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayerResult method), 75	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.InviteNotifyMessage method), 58	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayersActionDoneMessage method), 76	(pokerth-
FindInitializationErrors() proto.pokerth_pb2.InvitePlayerToGameMessage method), 59	(pokerth-	FindInitializationErrors() proto.pokerth_pb2.PlayersTurnMessage method), 78	(pokerth-

FindInitializationErrors() (proto.pokerth_pb2.PokerTHMessage method), 79	(pokerth-	FindInitializationErrors() (proto.pokerth_pb2.UnknownAvatarMessage method), 103	(pokerth-
FindInitializationErrors() (proto.pokerth_pb2.RejectGameInvitationMessage method), 88	(pokerth-	FindInitializationErrors() (proto.pokerth_pb2.VoteKickReplyMessage method), 104	(pokerth-
FindInitializationErrors() (proto.pokerth_pb2.RejectInvNotifyMessage method), 89	(pokerth-	FindInitializationErrors() (proto.pokerth_pb2.VoteKickRequestMessage method), 105	(pokerth-
FindInitializationErrors() (proto.pokerth_pb2.RejoinExistingGameMessage method), 90	(pokerth-	FindInitializationErrors() (proto.pokerth_pb2>YourActionRejectedMessage method), 106	(pokerth-
FindInitializationErrors() (proto.pokerth_pb2.RemovedFromGameMessage method), 91	(pokerth-	firstSmallBlind (pokerthproto.lobby.GameInfo attribute), 9	
FindInitializationErrors() (proto.pokerth_pb2.ReportAvatarAckMessage method), 92	(pokerth-	firstSmallBlind (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69	
FindInitializationErrors() (proto.pokerth_pb2.ReportAvatarMessage method), 93	(pokerth-	FIRSTSMALLBLIND_FIELD_NUMBER (pokerth- proto.pokerth_pb2.NetGameInfo attribute), 68	
FindInitializationErrors() (proto.pokerth_pb2.ReportGameAckMessage method), 94	(pokerth-	FLOP (pokerthproto.poker.Round attribute), 11	
FindInitializationErrors() (proto.pokerth_pb2.ReportGameMessage method), 94	(pokerth-	flopCard1 (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.ResetTimeoutMessage method), 95	(pokerth-	FLOPCARD1_FIELD_NUMBER (pokerth- proto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.ShowMyCardsRequestMessage method), 96	(pokerth-	flopCard2 (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.StartEventAckMessage method), 97	(pokerth-	FLOPCARD2_FIELD_NUMBER (pokerth- proto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.StartEventMessage method), 98	(pokerth-	flopCard3 (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.StartKickPetitionMessage method), 99	(pokerth-	FLOPCARD3_FIELD_NUMBER (pokerth- proto.pokerth_pb2.DealFlopCardsMessage attribute), 31	
FindInitializationErrors() (proto.pokerth_pb2.StatisticsMessage method), 100	(pokerth-	FOLD (pokerthproto.poker.Action attribute), 11	
FindInitializationErrors() (proto.pokerth_pb2.StatisticsMessage.StatisticsData method), 100	(pokerth-	FromString() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage static method), 12	
FindInitializationErrors() (proto.pokerth_pb2.SubscriptionRequestMessage method), 101	(pokerth-	FromString() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage static method), 13	
FindInitializationErrors() (proto.pokerth_pb2.TimeoutWarningMessage method), 102	(pokerth-	FromString() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage static method), 14	
		FromString() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage static method), 15	
		FromString() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage static method), 15	
		FromString() (pokerthproto.pokerth_pb2.AllInShowCardsMessage static method), 16	
		FromString() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerA static method), 17	
		FromString() (pokerthproto.pokerth_pb2.AnnounceMessage static method), 18	
		FromString() (pokerthproto.pokerth_pb2.AnnounceMessage.Version static method), 19	

- FromString() (pokerthproto.pokerth_pb2.AskKickDeniedMessage static method), 20
- FromString() (pokerthproto.pokerth_pb2.AskKickPlayerMessage static method), 21
- FromString() (pokerthproto.pokerth_pb2.AuthClientResponseMessage static method), 22
- FromString() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage static method), 23
- FromString() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage static method), 24
- FromString() (pokerthproto.pokerth_pb2.AvatarDataMessage static method), 24
- FromString() (pokerthproto.pokerth_pb2.AvatarEndMessage static method), 25
- FromString() (pokerthproto.pokerth_pb2.AvatarHeaderMessage static method), 26
- FromString() (pokerthproto.pokerth_pb2.AvatarRequestMessage static method), 27
- FromString() (pokerthproto.pokerth_pb2.ChatMessage static method), 28
- FromString() (pokerthproto.pokerth_pb2.ChatRejectMessage static method), 29
- FromString() (pokerthproto.pokerth_pb2.ChatRequestMessage static method), 30
- FromString() (pokerthproto.pokerth_pb2.DealFlopCardsMessage static method), 31
- FromString() (pokerthproto.pokerth_pb2.DealRiverCardMessage static method), 32
- FromString() (pokerthproto.pokerth_pb2.DealTurnCardMessage static method), 33
- FromString() (pokerthproto.pokerth_pb2.DialogMessage static method), 33
- FromString() (pokerthproto.pokerth_pb2.EndKickPetitionMessage static method), 34
- FromString() (pokerthproto.pokerth_pb2.EndOfGameMessage static method), 36
- FromString() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage static method), 36
- FromString() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage static method), 37
- FromString() (pokerthproto.pokerth_pb2.ErrorMessage static method), 38
- FromString() (pokerthproto.pokerth_pb2.GameAdminChangeMessage static method), 40
- FromString() (pokerthproto.pokerth_pb2.GameListAdminChangeMessage static method), 40
- FromString() (pokerthproto.pokerth_pb2.GameListNewMessage static method), 41
- FromString() (pokerthproto.pokerth_pb2.GameListPlayerJoinMessage static method), 43
- FromString() (pokerthproto.pokerth_pb2.GameListPlayerLeaveMessage static method), 43
- FromString() (pokerthproto.pokerth_pb2.GameListSpectatorJoinMessage static method), 44
- FromString() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage static method), 45
- FromString() (pokerthproto.pokerth_pb2.GameListUpdateMessage static method), 46
- FromString() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage static method), 47
- FromString() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage static method), 48
- FromString() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage static method), 49
- FromString() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage static method), 50
- FromString() (pokerthproto.pokerth_pb2.GameStartInitialMessage static method), 51
- FromString() (pokerthproto.pokerth_pb2.GameStartRejoinMessage static method), 52
- FromString() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinInfo static method), 52
- FromString() (pokerthproto.pokerth_pb2.HandStartMessage static method), 53
- FromString() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards static method), 54
- FromString() (pokerthproto.pokerth_pb2.InitAckMessage static method), 55
- FromString() (pokerthproto.pokerth_pb2.InitMessage static method), 57
- FromString() (pokerthproto.pokerth_pb2.InviteNotifyMessage static method), 58
- FromString() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage static method), 59
- FromString() (pokerthproto.pokerth_pb2.JoinExistingGameMessage static method), 60
- FromString() (pokerthproto.pokerth_pb2.JoinGameAckMessage static method), 61
- FromString() (pokerthproto.pokerth_pb2.JoinGameFailedMessage static method), 62
- FromString() (pokerthproto.pokerth_pb2.JoinNewGameMessage static method), 63
- FromString() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage static method), 64
- FromString() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage static method), 65
- FromString() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage static method), 66
- FromString() (pokerthproto.pokerth_pb2.MyActionRequestMessage static method), 67
- FromString() (pokerthproto.pokerth_pb2.NetGameInfo static method), 68
- FromString() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage static method), 70
- FromString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage static method), 71
- FromString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfo static method), 72

- FromString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage static method), 71
- FromString() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage static method), 74
- FromString() (pokerthproto.pokerth_pb2.PlayerListMessage static method), 74
- FromString() (pokerthproto.pokerth_pb2.PlayerResult static method), 75
- FromString() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage static method), 76
- FromString() (pokerthproto.pokerth_pb2.PlayersTurnMessage static method), 78
- FromString() (pokerthproto.pokerth_pb2.PokerTHMessage static method), 80
- FromString() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage static method), 88
- FromString() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage static method), 89
- FromString() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage static method), 90
- FromString() (pokerthproto.pokerth_pb2.RemovedFromGameMessage static method), 91
- FromString() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage static method), 92
- FromString() (pokerthproto.pokerth_pb2.ReportAvatarMessage static method), 93
- FromString() (pokerthproto.pokerth_pb2.ReportGameAckMessage static method), 94
- FromString() (pokerthproto.pokerth_pb2.ReportGameMessage static method), 95
- FromString() (pokerthproto.pokerth_pb2.ResetTimeoutMessage static method), 95
- FromString() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage static method), 96
- FromString() (pokerthproto.pokerth_pb2.StartEventAckMessage static method), 97
- FromString() (pokerthproto.pokerth_pb2.StartEventMessage static method), 98
- FromString() (pokerthproto.pokerth_pb2.StartKickPetitionMessage static method), 99
- FromString() (pokerthproto.pokerth_pb2.StatisticsMessage static method), 100
- FromString() (pokerthproto.pokerth_pb2.StatisticsMessage static method), 100
- FromString() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage static method), 101
- FromString() (pokerthproto.pokerth_pb2.TimeoutWarningMessage static method), 102
- FromString() (pokerthproto.pokerth_pb2.UnknownAvatarMessage static method), 103
- FromString() (pokerthproto.pokerth_pb2.VoteKickReplyMessage static method), 104
- FromString() (pokerthproto.pokerth_pb2.VoteKickRequestMessage static method), 105
- FromString() (pokerthproto.pokerth_pb2>YourActionRejectedMessage static method), 106
- G**
- Game (class in pokerthproto.game), 7
- GAME_JOINED (pokerthproto.protocol.States attribute), 112
- GAME_STARTED (pokerthproto.protocol.States attribute), 112
- GameAdminChangedMessage (class in pokerthproto.pokerth_pb2), 39
- gameAdminChangedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85
- GAMEADMINCHANGEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- gameAdminChangedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110
- gameUsed (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
- GameError, 9
- gameId (pokerthproto.game.Game attribute), 8
- gameId (pokerthproto.lobby.GameInfo attribute), 9
- gameId (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 18
- gameId (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- gameId (pokerthproto.pokerth_pb2.AskKickPlayerMessage attribute), 22
- gameId (pokerthproto.pokerth_pb2.ChatMessage attribute), 29
- gameId (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31
- gameId (pokerthproto.pokerth_pb2.DealRiverCardMessage attribute), 32
- gameId (pokerthproto.pokerth_pb2.DealTurnCardMessage attribute), 33
- gameId (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- gameId (pokerthproto.pokerth_pb2.EndOfGameMessage attribute), 36
- gameId (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37
- gameId (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage attribute), 38
- gameId (pokerthproto.pokerth_pb2.GameAdminChangedMessage attribute), 40
- gameId (pokerthproto.pokerth_pb2.GameListAdminChangedMessage attribute), 41
- gameId (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42

gameId (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage attribute), 43

gameId (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage attribute), 44

gameId (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage attribute), 45

gameId (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage attribute), 46

gameId (pokerthproto.pokerth_pb2.GameListUpdateMessage attribute), 46

gameId (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47

gameId (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48

gameId (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage attribute), 49

gameId (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50

gameId (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51

gameId (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53

gameId (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55

gameId (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58

gameId (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage attribute), 59

gameId (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60

gameId (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61

gameId (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62

gameId (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64

gameId (pokerthproto.pokerth_pb2.KickPlayerRequestMessage attribute), 65

gameId (pokerthproto.pokerth_pb2.LeaveGameRequestMessage attribute), 66

gameId (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67

gameId (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77

gameId (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78

gameId (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 88

gameId (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 89

gameId (pokerthproto.pokerth_pb2.RejoinExistingGameMessage attribute), 90

gameId (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91

gameId (pokerthproto.pokerth_pb2.StartEventAckMessage attribute), 97

gameId (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98

gameId (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99

gameId (pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105

gameId (pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 106

gameId (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 107

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 16

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 20

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AskKickPlayerMessage attribute), 21

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ChatMessage attribute), 28

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.DealFlopCardsMessage attribute), 31

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.DealRiverCardMessage attribute), 32

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.DealTurnCardMessage attribute), 33

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 34

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfGameMessage attribute), 36

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 36

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage attribute), 37

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameAdminChangedMessage attribute), 40

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListAdminChangedMessage attribute), 40

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 41

GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage attribute), 43	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPlayerRequestMessage attribute), 65
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage attribute), 43	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.LeaveGameRequestMessage attribute), 66
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage attribute), 44	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage attribute), 45	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListUpdateMessage attribute), 46	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 88
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 89
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage attribute), 49	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejoinExistingGameMessage attribute), 90
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartEventAckMessage attribute), 97
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 52	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 54	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 104
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage attribute), 59	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 105
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60	GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 106
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61	GameInfo (class in pokerthproto.lobby), 9
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62	gameInfo (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42
GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64	gameInfo (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61
	gameInfo (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 64
	GAMEINFO_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 64

attribute), 41

GAMEINFO_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61

GAMEINFO_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 63

gameInfos (pokerthproto.lobby.Lobby attribute), 10

gameIsFull (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62

gameIsFull (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91

gameIsRunning (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62

gameIsRunning (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91

GameListAdminChangedMessage (class in pokerthproto.pokerth_pb2), 40

gameListAdminChangedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTADMINCHANGEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListAdminChangedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListNewMessage (class in pokerthproto.pokerth_pb2), 41

gameListNewMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTNEWMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListNewReceived() (pokerthproto.protocol.ClientProtocol method), 108

gameListNewReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListPlayerJoinedMessage (class in pokerthproto.pokerth_pb2), 42

gameListPlayerJoinedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTPLAYERJOINEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListPlayerJoinedReceived() (pokerthproto.protocol.ClientProtocol method), 108

gameListPlayerJoinedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListPlayerLeftMessage (class in pokerthproto.pokerth_pb2), 43

gameListPlayerLeftMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTPLAYERLEFTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListPlayerLeftReceived() (pokerthproto.protocol.ClientProtocol method), 108

gameListPlayerLeftReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListSpectatorJoinedMessage (class in pokerthproto.pokerth_pb2), 44

gameListSpectatorJoinedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTSPECTATORJOINEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListSpectatorJoinedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListSpectatorLeftMessage (class in pokerthproto.pokerth_pb2), 45

gameListSpectatorLeftMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTSPECTATORLEFTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListSpectatorLeftReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

GameListUpdateMessage (class in pokerthproto.pokerth_pb2), 46

gameListUpdateMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85

GAMELISTUPDATEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

gameListUpdateReceived() (pokerthproto.protocol.ClientProtocol method), 108

gameListUpdateReceived() (pokerthproto.protocol.PokerTHProtocol method), 110

gameMode (pokerthproto.lobby.GameInfo attribute), 9

gameMode (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42

gameMode (pokerthproto.pokerth_pb2.GameListUpdateMessage attribute), 47

GAMEMODE_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 41	gameRemoveInvalid (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14
GAMEMODE_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListUpdateMessage attribute), 46	gameReportAccepted (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94
gameName (pokerthproto.lobby.GameInfo attribute), 9	gameReportDuplicate (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94
gameName (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69	gameReportInvalid (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94
GAMENAME_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68	GameSpectatorJoinedMessage (class in pokerthproto.pokerth_pb2), 49
gameNameInUse (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62	gameSpectatorJoinedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
GamePlayerJoinedMessage (class in pokerthproto.pokerth_pb2), 47	GAMESPECTATORJOINEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
gamePlayerJoinedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 85	gameSpectatorJoinedReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
GAMEPLAYERJOINEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	GameSpectatorLeftMessage (class in pokerthproto.pokerth_pb2), 49
gamePlayerJoinedReceived() (pokerthproto.protocol.ClientProtocol method), 108	gameSpectatorLeftMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
gamePlayerJoinedReceived() (pokerthproto.protocol.PokerTHProtocol method), 110	GAMESPECTATORLEFTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
GamePlayerLeftMessage (class in pokerthproto.pokerth_pb2), 48	gameSpectatorLeftReason (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50
gamePlayerLeftMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86	GAMESPECTATORLEFTREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50
GAMEPLAYERLEFTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	gameSpectatorLeftReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
GamePlayerLeftReason (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	GameStartInitialMessage (class in pokerthproto.pokerth_pb2), 50
gamePlayerLeftReason (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	gameStartInitialMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
GAMEPLAYERLEFTREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	GAMESTARTINITIALMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
gamePlayerLeftReceived() (pokerthproto.protocol.ClientProtocol method), 108	gameStartInitialReceived() (pokerthproto.protocol.ClientProtocol method), 108
gamePlayerLeftReceived() (pokerthproto.protocol.PokerTHProtocol method), 110	gameStartInitialReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
gameRemoveAccepted (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14	

GameStartRejoinMessage (class in pokerthproto.pokerth_pb2), 51
 gameStartRejoinMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
 GameStartRejoinMessage.RejoinPlayerData (class in pokerthproto.pokerth_pb2), 52
 GAMESTARTREJOINMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
 gameStartRejoinReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 gameState (pokerthproto.game.RoundInfo attribute), 9
 gameState (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
 gameState (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
 gameState (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78
 gameState (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 107
 GAMESTATE_FIELD_NUMBER (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
 GAMESTATE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
 GAMESTATE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78
 GAMESTATE_FIELD_NUMBER (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 106
 gameTimeout (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
 getActions() (pokerthproto.game.Game method), 8
 getGameInfo() (pokerthproto.lobby.Lobby method), 10
 getGameInfoId() (pokerthproto.lobby.Lobby method), 10
 getMsg() (pokerthproto.lobby.GameInfo method), 10
 getPlayer() (pokerthproto.game.Game method), 8
 getPlayer() (pokerthproto.lobby.Lobby method), 10
 guestLogin (pokerthproto.pokerth_pb2.InitMessage attribute), 57

H

handleChat() (pokerthproto.protocol.ClientProtocol method), 108
 handleEndOfGame() (pokerthproto.protocol.ClientProtocol method), 108
 handleEndOfHand() (pokerthproto.protocol.ClientProtocol method), 108
 handleInsideLobby() (pokerthproto.protocol.ClientProtocol method), 108
 handleMyTurn() (pokerthproto.protocol.ClientProtocol method), 108
 handleOthersTurn() (pokerthproto.protocol.ClientProtocol method), 108
 handNum (pokerthproto.game.Game attribute), 8
 handNum (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53
 handNum (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
 HANDNUM_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 52
 HANDNUM_FIELD_NUMBER (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
 HandStartMessage (class in pokerthproto.pokerth_pb2), 53
 handStartMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
 HandStartMessage.PlainCards (class in pokerthproto.pokerth_pb2), 54
 HANDSTARTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
 handStartReceived() (pokerthproto.protocol.ClientProtocol method), 108
 handStartReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 HasField() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12
 HasField() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13
 HasField() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14
 HasField() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15
 HasField() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 15
 HasField() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16
 HasField() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17
 HasField() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18
 HasField() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19
 HasField() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20
 HasField() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21

HasField() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22	HasField() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47
HasField() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23	HasField() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48
HasField() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 24	HasField() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49
HasField() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 24	HasField() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50
HasField() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25	HasField() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 51
HasField() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26	HasField() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 52
HasField() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27	HasField() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayer method), 52
HasField() (pokerthproto.pokerth_pb2.ChatMessage method), 28	HasField() (pokerthproto.pokerth_pb2.HandStartMessage method), 54
HasField() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29	HasField() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54
HasField() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30	HasField() (pokerthproto.pokerth_pb2.InitAckMessage method), 55
HasField() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31	HasField() (pokerthproto.pokerth_pb2.InitMessage method), 57
HasField() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32	HasField() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58
HasField() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33	HasField() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59
HasField() (pokerthproto.pokerth_pb2.DialogMessage method), 34	HasField() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60
HasField() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34	HasField() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61
HasField() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 36	HasField() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62
HasField() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 36	HasField() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63
HasField() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 37	HasField() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64
HasField() (pokerthproto.pokerth_pb2.ErrorMessage method), 38	HasField() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65
HasField() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 40	HasField() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
HasField() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 40	HasField() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67
HasField() (pokerthproto.pokerth_pb2.GameListNewMessage method), 41	HasField() (pokerthproto.pokerth_pb2.NetGameInfo method), 68
HasField() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	HasField() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70
HasField() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 43	HasField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71
HasField() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44	HasField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoReply method), 72
HasField() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	HasField() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoReply method), 71
HasField() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46	HasField() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74

HasField() (pokerthproto.pokerth_pb2.PlayerListMessage method), 74
 HasField() (pokerthproto.pokerth_pb2.PlayerResult method), 75
 HasField() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77
 HasField() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78
 HasField() (pokerthproto.pokerth_pb2.PokerTHMessage method), 80
 HasField() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88
 HasField() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89
 HasField() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90
 HasField() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91
 HasField() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92
 HasField() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93
 HasField() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94
 HasField() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95
 HasField() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 95
 HasField() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96
 HasField() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97
 HasField() (pokerthproto.pokerth_pb2.StartEventMessage method), 98
 HasField() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 99
 HasField() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100
 HasField() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 100
 HasField() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 101
 HasField() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 102
 HasField() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 103
 HasField() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104
 HasField() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105
 HasField() (pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 106
 highestSet (pokerthproto.game.Game attribute), 8
 highestSet (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
 HIGHESTSET_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
 INIT (pokerthproto.protocol.States attribute), 112
 InitAckMessage (class in pokerthproto.pokerth_pb2), 55
 initAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
 INITACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
 initAckReceived() (pokerthproto.protocol.ClientProtocol method), 109
 initAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 initAuthFailure (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 initBlocked (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 initInvalidPlayerName (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 InitMessage (class in pokerthproto.pokerth_pb2), 56
 initMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
 INITMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
 initPlayerNameInUse (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 initReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 initServerFull (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 initServerMaintenance (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 initVersionNotSupported (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 getToCard() (in module pokerthproto.poker), 11
 invalidGame (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62
 invalidPacket (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
 invalidPassword (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62

invalidSettings (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62	IsInitialized() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15
invalidState (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39	IsInitialized() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 15
InviteNotifyMessage (class in pokerthproto.pokerth_pb2), 58	IsInitialized() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16
inviteNotifyMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86	IsInitialized() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17
INVITENOTIFYMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	IsInitialized() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18
inviteNotifyReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	IsInitialized() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19
inviteOnlyGame (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69	IsInitialized() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20
InvitePlayerToGameMessage (class in pokerthproto.pokerth_pb2), 59	IsInitialized() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21
invitePlayerToGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86	IsInitialized() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22
INVITEPLAYERTOGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	IsInitialized() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23
invitePlayerToGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	IsInitialized() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 24
ipAddressBlocked (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62	IsInitialized() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 24
isGameAdmin (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47	IsInitialized() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25
ISGAMEADMIN_FIELD_NUMBER (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47	IsInitialized() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26
isHuman (pokerthproto.player.Player attribute), 11	IsInitialized() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27
isHuman (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 73	IsInitialized() (pokerthproto.pokerth_pb2.ChatMessage method), 28
ISHUMAN_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 72	IsInitialized() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
IsInitialized() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12	IsInitialized() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
IsInitialized() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13	
IsInitialized() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14	

IsInitialized() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31	IsInitialized() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32	IsInitialized() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33	IsInitialized() (pokerthproto.pokerth_pb2.DialogMessage method), 34	IsInitialized() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 34	IsInitialized() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 36	IsInitialized() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 37	IsInitialized() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 38	IsInitialized() (pokerthproto.pokerth_pb2.ErrorMessage method), 38	IsInitialized() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 40	IsInitialized() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 40	IsInitialized() (pokerthproto.pokerth_pb2.GameListNewMessage method), 41	IsInitialized() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	IsInitialized() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 43	IsInitialized() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44	IsInitialized() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	IsInitialized() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46	IsInitialized() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47	IsInitialized() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48	IsInitialized() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	IsInitialized() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50	IsInitialized() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 51	IsInitialized() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 52	IsInitialized() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 52	IsInitialized() (pokerthproto.pokerth_pb2.HandStartMessage method), 54	IsInitialized() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54	IsInitialized() (pokerthproto.pokerth_pb2.InitAckMessage method), 56	IsInitialized() (pokerthproto.pokerth_pb2.InitMessage method), 57	IsInitialized() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58	IsInitialized() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59	IsInitialized() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60	IsInitialized() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61	IsInitialized() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62	IsInitialized() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63	IsInitialized() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64	IsInitialized() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65	IsInitialized() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
--	--	---	---	--	--	---	---	--	---	---	--	---	---	--	--	---	---	---	--	--	---	--	---	--	---	--	---	---	---	---	--	---	--	---	--	---

IsInitialized() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67

IsInitialized() (pokerthproto.pokerth_pb2.NetGameInfo method), 68

IsInitialized() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70

IsInitialized() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71

IsInitialized() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 72

IsInitialized() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.AvatarData method), 72

IsInitialized() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74

IsInitialized() (pokerthproto.pokerth_pb2.PlayerListMessage method), 74

IsInitialized() (pokerthproto.pokerth_pb2.PlayerResult method), 75

IsInitialized() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77

IsInitialized() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78

IsInitialized() (pokerthproto.pokerth_pb2.PokerTHMessage method), 80

IsInitialized() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88

IsInitialized() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89

IsInitialized() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90

IsInitialized() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91

IsInitialized() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92

IsInitialized() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93

IsInitialized() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94

IsInitialized() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95

IsInitialized() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 95

IsInitialized() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96

IsInitialized() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97

IsInitialized() (pokerthproto.pokerth_pb2.StartEventMessage method), 98

IsInitialized() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 99

IsInitialized() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100

IsInitialized() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101

IsInitialized() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 102

IsInitialized() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 102

IsInitialized() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 103

IsInitialized() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104

IsInitialized() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105

IsInitialized() (pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 106

isPrivate (pokerthproto.lobby.GameInfo attribute), 10

isPrivate (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42

ISPRIVATE_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 41

J

JoinExistingGameMessage (class in pokerthproto.pokerth_pb2), 59

- joinExistingGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
- JOINEXISTINGGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- joinExistingGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- JoinGameAckMessage (class in pokerthproto.pokerth_pb2), 60
- joinGameAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
- JOINGAMEACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- joinGameAckReceived() (pokerthproto.protocol.ClientProtocol method), 109
- joinGameAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- JoinGameFailedMessage (class in pokerthproto.pokerth_pb2), 61
- joinGameFailedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
- JOINGAMEFAILEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- joinGameFailedReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- JoinGameFailureReason (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62
- joinGameFailureReason (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 63
- JOINGAMEFAILUREREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 62
- JoinNewGameMessage (class in pokerthproto.pokerth_pb2), 63
- joinNewGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
- JOINNEWGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- joinNewGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- K**
- keepLastBlind (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
- kickDeniedAlreadyInProgress (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- kickDeniedInvalidGameState (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- kickDeniedInvalidPlayerId (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- kickDeniedNotPossible (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- KickDeniedReason (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 20
- kickDeniedReason (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- KICKDENIEDREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 20
- kickDeniedTryAgainLater (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- kickedFromGame (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
- kickedFromServer (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
- KickPetitionUpdateMessage (class in pokerthproto.pokerth_pb2), 64
- kickPetitionUpdateMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
- KICKPETITIONUPDATEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80
- kickPetitionUpdateReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- kickPlayerId (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- KICKPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- KickPlayerRequestMessage (class in pokerthproto.pokerth_pb2), 65
- kickPlayerRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86

KICKPLAYERREQUESTMESSAGE_FIELD_NUMBER	ListFields() (pokerthproto.pokerth_pb2.AllInShowCardsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	ListFields() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16
kickPlayerRequestReceived()	(pokerthproto.protocol.PokerTHProtocol method), 111	ListFields() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAll method), 17
kickTimeoutSec	(pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99	ListFields() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18
KICKTIMEOUTSEC_FIELD_NUMBER	(pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99	ListFields() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19
		ListFields() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20
		ListFields() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21
		ListFields() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22
L		ListFields() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23
latestBetaRevision	(pokerthproto.pokerth_pb2.AnnounceMessage attribute), 19	ListFields() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 24
LATESTBETAREVISION_FIELD_NUMBER	(pokerthproto.pokerth_pb2.AnnounceMessage attribute), 18	ListFields() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 25
latestGameVersion	(pokerthproto.pokerth_pb2.AnnounceMessage attribute), 19	ListFields() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25
LATESTGAMEVERSION_FIELD_NUMBER	(pokerthproto.pokerth_pb2.AnnounceMessage attribute), 18	ListFields() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26
LeaveGameRequestMessage	(class in pokerthproto.pokerth_pb2), 66	ListFields() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27
leaveGameRequestMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86	ListFields() (pokerthproto.pokerth_pb2.ChatMessage method), 28
LEAVEGAMEREQUESTMESSAGE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80	ListFields() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
leaveGameRequestReceived()	(pokerthproto.protocol.PokerTHProtocol method), 111	ListFields() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
leftError	(pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	ListFields() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31
leftKicked	(pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48	ListFields() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32
leftOnRequest	(pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 49	ListFields() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33
ListFields() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12		ListFields() (pokerthproto.pokerth_pb2.DialogMessage method), 34
ListFields() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13		ListFields() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 35
ListFields() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14		ListFields() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 36
ListFields() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15		ListFields() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 37
ListFields() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 16		ListFields() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 38
		ListFields() (pokerthproto.pokerth_pb2.ErrorMessage method), 39
		ListFields() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 40
		ListFields() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 41

- ListFields() (pokerthproto.pokerth_pb2.GameListNewMessage method), 42
- ListFields() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 43
- ListFields() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 44
- ListFields() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 44
- ListFields() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45
- ListFields() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46
- ListFields() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47
- ListFields() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48
- ListFields() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49
- ListFields() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50
- ListFields() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 51
- ListFields() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 52
- ListFields() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.PlayerData method), 52
- ListFields() (pokerthproto.pokerth_pb2.HandStartMessage method), 54
- ListFields() (pokerthproto.pokerth_pb2.HandStartMessage.HandCards method), 54
- ListFields() (pokerthproto.pokerth_pb2.InitAckMessage method), 56
- ListFields() (pokerthproto.pokerth_pb2.InitMessage method), 57
- ListFields() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58
- ListFields() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59
- ListFields() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60
- ListFields() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61
- ListFields() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62
- ListFields() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63
- ListFields() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64
- ListFields() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65
- ListFields() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
- ListFields() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67
- ListFields() (pokerthproto.pokerth_pb2.NetGameInfo method), 68
- ListFields() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70
- ListFields() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71
- ListFields() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfo method), 72
- ListFields() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfo method), 72
- ListFields() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74
- ListFields() (pokerthproto.pokerth_pb2.PlayerListMessage method), 75
- ListFields() (pokerthproto.pokerth_pb2.PlayerResult method), 76
- ListFields() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77
- ListFields() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78
- ListFields() (pokerthproto.pokerth_pb2.PokerTHMessage method), 80
- ListFields() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88
- ListFields() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89
- ListFields() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90
- ListFields() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91
- ListFields() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92
- ListFields() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93
- ListFields() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94
- ListFields() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95
- ListFields() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 96
- ListFields() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96
- ListFields() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97
- ListFields() (pokerthproto.pokerth_pb2.StartEventMessage method), 98
- ListFields() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 99
- ListFields() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100
- ListFields() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101
- ListFields() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 102

ListFields() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 103	MergeFrom() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15
ListFields() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 104	MergeFrom() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 16
ListFields() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104	MergeFrom() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 16
ListFields() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105	MergeFrom() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17
ListFields() (pokerthproto.pokerth_pb2.YourActionRejectedMessage method), 106	MergeFrom() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18
Lobby (class in pokerthproto.lobby), 10	MergeFrom() (pokerthproto.pokerth_pb2.AnnounceMessage.Version method), 19
LOBBY (pokerthproto.protocol.States attribute), 112	MergeFrom() (pokerthproto.pokerth_pb2.AskKickDeniedMessage method), 20
LobbyError, 10	MergeFrom() (pokerthproto.pokerth_pb2.AskKickPlayerMessage method), 21
login (pokerthproto.pokerth_pb2.InitMessage attribute), 57	MergeFrom() (pokerthproto.pokerth_pb2.AuthClientResponseMessage method), 22
LOGIN_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute), 57	MergeFrom() (pokerthproto.pokerth_pb2.AuthServerChallengeMessage method), 23
LoginType (pokerthproto.pokerth_pb2.InitMessage attribute), 57	MergeFrom() (pokerthproto.pokerth_pb2.AuthServerVerificationMessage method), 24
M	MergeFrom() (pokerthproto.pokerth_pb2.AvatarDataMessage method), 25
majorVersion (pokerthproto.pokerth_pb2.AnnounceMessage.Version attribute), 19	MergeFrom() (pokerthproto.pokerth_pb2.AvatarEndMessage method), 25
MAJORVERSION_FIELD_NUMBER (pokerthproto.pokerth_pb2.AnnounceMessage.Version attribute), 19	MergeFrom() (pokerthproto.pokerth_pb2.AvatarHeaderMessage method), 26
makeSizeBytes() (in module pokerthproto.transport), 113	MergeFrom() (pokerthproto.pokerth_pb2.AvatarRequestMessage method), 27
manualBlinds (pokerthproto.lobby.GameInfo attribute), 10	MergeFrom() (pokerthproto.pokerth_pb2.ChatMessage method), 28
manualBlinds (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69	MergeFrom() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
MANUALBLINDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68	MergeFrom() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
maxNumPlayers (pokerthproto.lobby.GameInfo attribute), 10	
maxNumPlayers (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69	
MAXNUMPLAYERS_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68	
MergeFrom() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12	
MergeFrom() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13	
MergeFrom() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14	

MergeFrom() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31	MergeFrom() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32	MergeFrom() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33	MergeFrom() (pokerthproto.pokerth_pb2.DialogMessage method), 34	MergeFrom() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 35	MergeFrom() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 36	MergeFrom() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 37	MergeFrom() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 38	MergeFrom() (pokerthproto.pokerth_pb2.ErrorMessage method), 39	MergeFrom() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 40	MergeFrom() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 41	MergeFrom() (pokerthproto.pokerth_pb2.GameListNewMessage method), 42	MergeFrom() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	MergeFrom() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 44	MergeFrom() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45	MergeFrom() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	MergeFrom() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46	MergeFrom() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47	MergeFrom() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48	MergeFrom() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	MergeFrom() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50	MergeFrom() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 51	MergeFrom() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 52	MergeFrom() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 52	MergeFrom() (pokerthproto.pokerth_pb2.HandStartMessage method), 54	MergeFrom() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54	MergeFrom() (pokerthproto.pokerth_pb2.InitAckMessage method), 56	MergeFrom() (pokerthproto.pokerth_pb2.InitMessage method), 57	MergeFrom() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58	MergeFrom() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59	MergeFrom() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60	MergeFrom() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61	MergeFrom() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62	MergeFrom() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63	MergeFrom() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64	MergeFrom() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65	MergeFrom() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
--	--	---	---	--	--	---	---	--	---	---	--	---	---	--	--	---	---	---	--	--	---	--	---	--	---	--	---	---	---	---	--	---	--	---	--	---

MergeFrom()	(pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67	MergeFrom()	(pokerthproto.pokerth_pb2.ReportGameMessage method), 95
MergeFrom()	(pokerthproto.pokerth_pb2.NetGameInfo method), 68	MergeFrom()	(pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 96
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70	MergeFrom()	(pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 71	MergeFrom()	(pokerthproto.pokerth_pb2.StartEventAckMessage method), 97
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 72	MergeFrom()	(pokerthproto.pokerth_pb2.StartEventMessage method), 98
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.AvatarData method), 72	MergeFrom()	(pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 99
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74	MergeFrom()	(pokerthproto.pokerth_pb2.StatisticsMessage method), 100
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerListMessage method), 75	MergeFrom()	(pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101
MergeFrom()	(pokerthproto.pokerth_pb2.PlayerResult method), 76	MergeFrom()	(pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 102
MergeFrom()	(pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77	MergeFrom()	(pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 103
MergeFrom()	(pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78	MergeFrom()	(pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 104
MergeFrom()	(pokerthproto.pokerth_pb2.PokerTHMessage method), 81	MergeFrom()	(pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104
MergeFrom()	(pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88	MergeFrom()	(pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105
MergeFrom()	(pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89	MergeFrom()	(pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 106
MergeFrom()	(pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90	MergeFromString()	(pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12
MergeFrom()	(pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91	MergeFromString()	(pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13
MergeFrom()	(pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92	MergeFromString()	(pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14
MergeFrom()	(pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93	MergeFromString()	(pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 14
MergeFrom()	(pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94		

method), 15		method), 31
MergeFromString() proto.pokerth_pb2.AfterHandShowCardsMessage method), 16	(pokerth-	MergeFromString() proto.pokerth_pb2.DealRiverCardMessage method), 32
MergeFromString() proto.pokerth_pb2.AllInShowCardsMessage method), 16	(pokerth-	MergeFromString() proto.pokerth_pb2.DealTurnCardMessage method), 33
MergeFromString() proto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17	(pokerth-	MergeFromString() proto.pokerth_pb2.DialogMessage 34
MergeFromString() proto.pokerth_pb2.AnnounceMessage method), 18	(pokerth-	MergeFromString() proto.pokerth_pb2.EndKickPetitionMessage method), 35
MergeFromString() proto.pokerth_pb2.AnnounceMessage.Version method), 19	(pokerth-	MergeFromString() proto.pokerth_pb2.EndOfGameMessage method), 36
MergeFromString() proto.pokerth_pb2.AskKickDeniedMessage method), 20	(pokerth-	MergeFromString() proto.pokerth_pb2.EndOfHandHideCardsMessage method), 37
MergeFromString() proto.pokerth_pb2.AskKickPlayerMessage method), 21	(pokerth-	MergeFromString() proto.pokerth_pb2.EndOfHandShowCardsMessage method), 38
MergeFromString() proto.pokerth_pb2.AuthClientResponseMessage method), 22	(pokerth-	MergeFromString() proto.pokerth_pb2.ErrorMessage 39
MergeFromString() proto.pokerth_pb2.AuthServerChallengeMessage method), 23	(pokerth-	MergeFromString() proto.pokerth_pb2.GameAdminChangedMessage method), 40
MergeFromString() proto.pokerth_pb2.AuthServerVerificationMessage method), 24	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListAdminChangedMessage method), 41
MergeFromString() proto.pokerth_pb2.AvatarDataMessage method), 25	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListNewMessage method), 42
MergeFromString() proto.pokerth_pb2.AvatarEndMessage method), 25	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListPlayerJoinedMessage method), 43
MergeFromString() proto.pokerth_pb2.AvatarHeaderMessage method), 26	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListPlayerLeftMessage method), 44
MergeFromString() proto.pokerth_pb2.AvatarRequestMessage method), 27	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45
MergeFromString() proto.pokerth_pb2.ChatMessage 28	(pokerth- method),	MergeFromString() proto.pokerth_pb2.GameListSpectatorLeftMessage method), 45
MergeFromString() proto.pokerth_pb2.ChatRejectMessage method), 29	(pokerth-	MergeFromString() proto.pokerth_pb2.GameListUpdateMessage method), 46
MergeFromString() proto.pokerth_pb2.ChatRequestMessage method), 30	(pokerth-	MergeFromString() proto.pokerth_pb2.GamePlayerJoinedMessage method), 47
MergeFromString() proto.pokerth_pb2.DealFlopCardsMessage	(pokerth-	MergeFromString() proto.pokerth_pb2.GamePlayerLeftMessage

method), 48		method), 66	
MergeFromString() proto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	(pokerth-	MergeFromString() proto.pokerth_pb2.MyActionRequestMessage method), 67	(pokerth-
MergeFromString() proto.pokerth_pb2.GameSpectatorLeftMessage method), 50	(pokerth-	MergeFromString() proto.pokerth_pb2.NetGameInfo 68	(pokerth-
MergeFromString() proto.pokerth_pb2.GameStartInitialMessage method), 51	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerIdChangedMessage method), 70	(pokerth-
MergeFromString() proto.pokerth_pb2.GameStartRejoinMessage method), 52	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerInfoReplyMessage method), 71	(pokerth-
MergeFromString() proto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 52	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 72	(pokerth-
MergeFromString() proto.pokerth_pb2.HandStartMessage 54	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.Avat method), 72	(pokerth-
MergeFromString() proto.pokerth_pb2.HandStartMessage.PlainCards method), 54	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerInfoRequestMessage method), 74	(pokerth-
MergeFromString() proto.pokerth_pb2.InitAckMessage 56	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerListMessage method), 75	(pokerth-
MergeFromString() proto.pokerth_pb2.InitMessage 57	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayerResult 76	(pokerth-
MergeFromString() proto.pokerth_pb2.InviteNotifyMessage method), 58	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayersActionDoneMessage method), 77	(pokerth-
MergeFromString() proto.pokerth_pb2.InvitePlayerToGameMessage method), 59	(pokerth-	MergeFromString() proto.pokerth_pb2.PlayersTurnMessage method), 78	(pokerth-
MergeFromString() proto.pokerth_pb2.JoinExistingGameMessage method), 60	(pokerth-	MergeFromString() proto.pokerth_pb2.PokerTHMessage 81	(pokerth-
MergeFromString() proto.pokerth_pb2.JoinGameAckMessage method), 61	(pokerth-	MergeFromString() proto.pokerth_pb2.RejectGameInvitationMessage method), 88	(pokerth-
MergeFromString() proto.pokerth_pb2.JoinGameFailedMessage method), 62	(pokerth-	MergeFromString() proto.pokerth_pb2.RejectInvNotifyMessage method), 89	(pokerth-
MergeFromString() proto.pokerth_pb2.JoinNewGameMessage method), 63	(pokerth-	MergeFromString() proto.pokerth_pb2.RejoinExistingGameMessage method), 90	(pokerth-
MergeFromString() proto.pokerth_pb2.KickPetitionUpdateMessage method), 64	(pokerth-	MergeFromString() proto.pokerth_pb2.RemovedFromGameMessage method), 91	(pokerth-
MergeFromString() proto.pokerth_pb2.KickPlayerRequestMessage method), 65	(pokerth-	MergeFromString() proto.pokerth_pb2.ReportAvatarAckMessage method), 92	(pokerth-
MergeFromString() proto.pokerth_pb2.LeaveGameRequestMessage	(pokerth-	MergeFromString() proto.pokerth_pb2.ReportAvatarMessage	(pokerth-

method), 93

MergeFromString() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94

MergeFromString() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95

MergeFromString() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 96

MergeFromString() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96

MergeFromString() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97

MergeFromString() (pokerthproto.pokerth_pb2.StartEventMessage method), 98

MergeFromString() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 99

MergeFromString() (pokerthproto.pokerth_pb2.StatisticsMessage method), 100

MergeFromString() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101

MergeFromString() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 102

MergeFromString() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 103

MergeFromString() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 104

MergeFromString() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 104

MergeFromString() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 105

MergeFromString() (pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 106

messageType (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86

MESSAGETYPE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 80

minimumRaise (pokerthproto.game.Game attribute), 8

minimumRaise (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77

MINIMUMRAISE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77

minorVersion (pokerthproto.pokerth_pb2.AnnounceMessage.Version attribute), 19

MINORVERSION_FIELD_NUMBER (pokerthproto.pokerth_pb2.AnnounceMessage.Version attribute), 19

money (pokerthproto.player.Player attribute), 11

moneyWon (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37

moneyWon (pokerthproto.pokerth_pb2.PlayerResult attribute), 76

MONEYWON_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37

MONEYWON_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerResult attribute), 76

myAction (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67

MYACTION_FIELD_NUMBER (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67

MyActionRequestMessage (class in pokerthproto.pokerth_pb2), 66

myActionRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86

MYACTIONREQUESTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81

myActionRequestReceived() (pokerthproto.protocol.PokerTHProtocol method), 111

myBet (pokerthproto.game.Game attribute), 8

myLastSessionId (pokerthproto.pokerth_pb2.InitMessage attribute), 57

MYLASTSESSIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute), 57

myRejectReason (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 89

MYREJECTREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 88

myRelativeBet (pokerthproto.pokerth_pb2.MyActionRequestMessage

- attribute), 67
- MYRELATIVEBET_FIELD_NUMBER (pokerthproto.pokerth_pb2.MyActionRequestMessage attribute), 67
- ## N
- name (pokerthproto.game.RoundInfo attribute), 9
- name (pokerthproto.player.Player attribute), 11
- NetGameInfo (class in pokerthproto.pokerth_pb2), 67
- netGameType (pokerthproto.lobby.GameInfo attribute), 10
- NetGameType (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68
- netGameType (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
- NETGAMETYPE_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68
- newAdminPlayerId (pokerthproto.pokerth_pb2.GameAdminChangedMessage attribute), 40
- newAdminPlayerId (pokerthproto.pokerth_pb2.GameListAdminChangedMessage attribute), 41
- NEWADMINPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameAdminChangedMessage attribute), 40
- NEWADMINPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListAdminChangedMessage attribute), 41
- newPlayerId (pokerthproto.pokerth_pb2.PlayerIdChangedMessage attribute), 70
- NEWPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerIdChangedMessage attribute), 70
- nickName (pokerthproto.pokerth_pb2.InitMessage attribute), 58
- NICKNAME_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute), 57
- NONE (pokerthproto.poker.Action attribute), 11
- normalGame (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
- noSpectatorsAllowed (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 63
- notAllowedAsGuest (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 63
- notificationText (pokerthproto.pokerth_pb2.DialogMessage attribute), 34
- NOTIFICATIONTEXT_FIELD_NUMBER (pokerthproto.pokerth_pb2.DialogMessage attribute), 34
- notInvited (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 63
- numPlayersOnServer (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 20
- NUMPLAYERSONSERVER_FIELD_NUMBER (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 18
- numVotesAgainstKicking (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- numVotesAgainstKicking (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 65
- NUMVOTESAGAINSTKICKING_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- NUMVOTESAGAINSTKICKING_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64
- numVotesInFavourOfKicking (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- numVotesInFavourOfKicking (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 65
- NUMVOTESINFAVOUROFKICKING_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- NUMVOTESINFAVOUROFKICKING_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64
- numVotesNeededToKick (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 65
- numVotesNeededToKick (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- NUMVOTESNEEDEDTOKICK_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64
- NUMVOTESNEEDEDTOKICK_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- ## O
- oldPlayerId (pokerthproto.pokerth_pb2.PlayerIdChangedMessage attribute), 70
- OLDPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerIdChangedMessage attribute), 70
- othersCards (pokerthproto.game.Game attribute), 8

P

- pack() (in module pokerthproto.transport), 113
- password (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60
- password (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 64
- PASSWORD_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60
- PASSWORD_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinNewGameMessage attribute), 63
- petitionEndEnoughVotes (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionEndPlayerLeft (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- PetitionEndReason (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionEndReason (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- PETITIONENDREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionEndTimeout (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionEndTooFewPlayers (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionId (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- petitionId (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 65
- petitionId (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- petitionId (pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105
- petitionId (pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 106
- PETITIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- PETITIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage attribute), 64
- PETITIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
- PETITIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 104
- PETITIONID_FIELD_NUMBER (pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 106
- plainCard1 (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards attribute), 55
- PLAINCARD1_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards attribute), 54
- plainCard2 (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards attribute), 55
- PLAINCARD2_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards attribute), 54
- plainCards (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
- PLAINCARDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 54
- Player (class in pokerthproto.player), 10
- playerAction (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
- PLAYERACTION_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
- playerActionTimeout (pokerthproto.lobby.GameInfo attribute), 10
- playerActionTimeout (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
- PLAYERACTIONTIMEOUT_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68
- playerId (pokerthproto.player.Player attribute), 11
- playerId (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn attribute), 17
- playerId (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 21
- playerId (pokerthproto.pokerth_pb2.AskKickPlayerMessage attribute), 22
- playerId (pokerthproto.pokerth_pb2.ChatMessage attribute), 29
- playerId (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37
- playerId (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage attribute), 43
- playerId (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage attribute), 44
- playerId (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage attribute), 45
- playerId (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage attribute), 46
- playerId (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47

playerId (pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 49	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.GamePlayerJoinedMessage attribute), 47
playerId (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage attribute), 49	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.GamePlayerLeftMessage attribute), 48
playerId (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage attribute), 49
playerId (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.GameSpectatorLeftMessage attribute), 50
playerId (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage attribute), 59	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData attribute), 52
playerId (pokerthproto.pokerth_pb2.KickPlayerRequestMessage attribute), 66	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.InvitePlayerToGameMessage attribute), 59
playerId (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 73	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.KickPlayerRequestMessage attribute), 65
playerId (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage attribute), 74	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 71
playerId (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayerInfoRequestMessage attribute), 74
playerId (pokerthproto.pokerth_pb2.PlayerResult attribute), 76	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerId attribute), 17
playerId (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75
playerId (pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 20
playerId (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 89	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayerResult attribute), 76
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerId attribute), 17	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AskKickDeniedMessage attribute), 20	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PlayersTurnMessage attribute), 78
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AskKickPlayerMessage attribute), 21	PLAYERID_FIELD_NUMBER	(pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 89
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ChatMessage attribute), 28	PLAYERID_FIELD_NUMBER	playerIdByWhom (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37	PLAYERID_FIELD_NUMBER	PLAYERIDBYWHOM_FIELD_NUMBER (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage attribute), 43	PLAYERID_FIELD_NUMBER	PlayerIdChangedMessage (class in pokerthproto.pokerth_pb2), 70
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage attribute), 44	PLAYERIDBYWHOM_FIELD_NUMBER	playerIdChangedMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage attribute), 45		
PLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage attribute), 45		

PLAYERIDCHANGEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	PlayerListMessage (class in pokerthproto.pokerth_pb2), 74
playerIdChangedReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	playerListMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
playerIds (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42	PLAYERLISTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
PLAYERIDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42	playerListNew (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75
playerIdWho (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 59	PlayerListNotification (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75
PLAYERIDWHO_FIELD_NUMBER (pokerthproto.pokerth_pb2.InviteNotifyMessage attribute), 58	playerListNotification (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75
playerInfoData (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 73	PLAYERLISTNOTIFICATION_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75
PLAYERINFODATA_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 71	playerListReceived() (pokerthproto.protocol.ClientProtocol method), 109
PlayerInfoReplyMessage (class in pokerthproto.pokerth_pb2), 70	playerListReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
playerInfoReplyMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 86	playerMoney (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37
PlayerInfoReplyMessage.PlayerInfoData (class in pokerthproto.pokerth_pb2), 71	playerMoney (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData attribute), 53
PlayerInfoReplyMessage.PlayerInfoData.AvatarData (class in pokerthproto.pokerth_pb2), 71	playerMoney (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
PLAYERINFOREPLYMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	playerMoney (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
playerInfoReplyReceived() (pokerthproto.protocol.ClientProtocol method), 109	PLAYERMONEY_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage attribute), 37
playerInfoReplyReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	PLAYERMONEY_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData attribute), 53
PlayerInfoRequestMessage (class in pokerthproto.pokerth_pb2), 73	PLAYERMONEY_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
playerInfoRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87	PLAYERMONEY_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayersActionDoneMessage attribute), 77
PLAYERINFOREQUESTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	playerName (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 73
playerInfoRequestReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	PLAYERNAME_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData attribute), 73
playerListLeft (pokerthproto.pokerth_pb2.PlayerListMessage attribute), 75	playerRejectReason (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 90

PLAYERREJECTREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.RejectInvNotifyMessage attribute), 81
PlayerResult (class in pokerthproto.pokerth_pb2), 75
playerResult (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage attribute), 16
PLAYERRESULT_FIELD_NUMBER (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage attribute), 16
playerResults (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage attribute), 38
PLAYERRESULTS_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage attribute), 38
playerRights (pokerthproto.player.Player attribute), 11
playerRights (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 73
PLAYERRIGHTS_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage attribute), 73
players (pokerthproto.game.Game attribute), 9
players (pokerthproto.lobby.GameInfo attribute), 10
players (pokerthproto.lobby.Lobby attribute), 10
PlayersActionDoneMessage (class in pokerthproto.pokerth_pb2), 76
playersActionDoneMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
PLAYERSACTIONDONEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
playersActionDoneReceived() (pokerthproto.protocol.ClientProtocol method), 109
playersActionDoneReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
playersAllIn (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 18
PLAYERSALLIN_FIELD_NUMBER (pokerthproto.pokerth_pb2.AllInShowCardsMessage attribute), 16
playerSeats (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51
PLAYERSEATS_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51
PlayersTurnMessage (class in pokerthproto.pokerth_pb2), 78
playersTurnMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
PLAYERSTURNMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
playersTurnReceived() (pokerthproto.protocol.ClientProtocol method), 109
playersTurnReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
pocketCards (pokerthproto.game.Game attribute), 9
PokerTHMessage (class in pokerthproto.pokerth_pb2), 78
PokerTHMessageType (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
pokerthproto (module), 113
pokerthproto.game (module), 7
pokerthproto.lobby (module), 9
pokerthproto.player (module), 10
pokerthproto.playerInfoData (module), 11
pokerthproto.pokerth_pb2 (module), 12
pokerthproto.protocol (module), 107
pokerthproto.proxy (module), 112
pokerthproto.transport (module), 113
PokerTHProtocol (class in pokerthproto.protocol), 109
PREFLOP (pokerthproto.poker.Round attribute), 11
proposedGuiSpeed (pokerthproto.lobby.GameInfo attribute), 10
proposedGuiSpeed (pokerthproto.pokerth_pb2.NetGameInfo attribute), 69
PROPOSEDGUI SPEED_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68
proposingPlayerId (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
PROPOSINGPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartKickPetitionMessage attribute), 99
protocol (pokerthproto.protocol.ClientProtocolFactory attribute), 109
protocol (pokerthproto.proxy.ClientProtocolFactory attribute), 112
protocol (pokerthproto.proxy.ProxyProtocolFactory attribute), 113
protocolVersion (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 20
PROTOCOLVERSION_FIELD_NUMBER (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 18
ProxyProtocol (class in pokerthproto.proxy), 112
ProxyProtocolFactory (class in pokerthproto.proxy), 112
R
RAISE (pokerthproto.poker.Action attribute), 11

raiseByEndValue proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AllInShowCardsMessage static method), 17	(pokerth- attribute),
raiseEveryHands (pokerthproto.lobby.GameInfo attribute), 10	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn static method), 17	(pokerth- attribute),
raiseEveryHands proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AnnounceMessage static method), 18	(pokerth- attribute),
RAISEEVERYHANDS_FIELD_NUMBER proto.pokerth_pb2.NetGameInfo 68	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AnnounceMessage.Version static method), 19	(pokerth- attribute),
raiseEveryMinutes proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AskKickDeniedMessage static method), 20	(pokerth- attribute),
RAISEEVERYMINUTES_FIELD_NUMBER proto.pokerth_pb2.NetGameInfo attribute), 68	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AskKickPlayerMessage static method), 21	(pokerth- attribute),
raiseIntervalMode (pokerthproto.lobby.GameInfo attribute), 10	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AuthClientResponseMessage static method), 22	(pokerth- attribute),
RaiseIntervalMode proto.pokerth_pb2.NetGameInfo 68	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AuthServerChallengeMessage static method), 23	(pokerth- attribute),
raiseIntervalMode proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AuthServerVerificationMessage static method), 24	(pokerth- attribute),
RAISEINTERVALMODE_FIELD_NUMBER proto.pokerth_pb2.NetGameInfo attribute), 68	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AvatarDataMessage static method), 25	(pokerth- attribute),
raiseOnHandNum proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AvatarEndMessage static method), 26	(pokerth- attribute),
raiseOnMinutes proto.pokerth_pb2.NetGameInfo 69	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AvatarHeaderMessage static method), 26	(pokerth- attribute),
rankingGame (pokerthproto.pokerth_pb2.NetGameInfo attribute), 70	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.AvatarRequestMessage static method), 27	(pokerth- attribute),
readSizeBytes() (in module pokerthproto.transport), 113	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.ChatMessage static method), 28	(pokerth- attribute),
registeredOnlyGame proto.pokerth_pb2.NetGameInfo 70	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.ChatRejectMessage static method), 29	(pokerth- attribute),
RegisterExtension() proto.pokerth_pb2.AdminBanPlayerAckMessage static method), 12	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.ChatRequestMessage static method), 30	(pokerth- attribute),
RegisterExtension() proto.pokerth_pb2.AdminBanPlayerMessage static method), 13	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.DealFlopCardsMessage static method), 31	(pokerth- attribute),
RegisterExtension() proto.pokerth_pb2.AdminRemoveGameAckMessage static method), 14	(pokerth- attribute),	RegisterExtension() proto.pokerth_pb2.DealRiverCardMessage static method), 32	(pokerth- attribute),
RegisterExtension() proto.pokerth_pb2.AdminRemoveGameMessage static method), 15	(pokerth- attribute),		
RegisterExtension() proto.pokerth_pb2.AfterHandShowCardsMessage static method), 16	(pokerth- attribute),		

RegisterExtension() proto.pokerth_pb2.DealTurnCardMessage static method), 33	(pokerth-	RegisterExtension() proto.pokerth_pb2.GameSpectatorLeftMessage static method), 50	(pokerth-
RegisterExtension() proto.pokerth_pb2.DialogMessage method), 34	static	RegisterExtension() proto.pokerth_pb2.GameStartInitialMessage static method), 51	(pokerth-
RegisterExtension() proto.pokerth_pb2.EndKickPetitionMessage static method), 35	(pokerth-	RegisterExtension() proto.pokerth_pb2.GameStartRejoinMessage static method), 52	(pokerth-
RegisterExtension() proto.pokerth_pb2.EndOfGameMessage static method), 36	(pokerth-	RegisterExtension() proto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData static method), 53	(pokerth-
RegisterExtension() proto.pokerth_pb2.EndOfHandHideCardsMessage static method), 37	(pokerth-	RegisterExtension() proto.pokerth_pb2.HandStartMessage method), 55	static
RegisterExtension() proto.pokerth_pb2.EndOfHandShowCardsMessage static method), 38	(pokerth-	RegisterExtension() proto.pokerth_pb2.HandStartMessage.PlainCards static method), 54	(pokerth-
RegisterExtension() proto.pokerth_pb2.ErrorMessage method), 39	static	RegisterExtension() proto.pokerth_pb2.InitAckMessage method), 56	static
RegisterExtension() proto.pokerth_pb2.GameAdminChangedMessage static method), 40	(pokerth-	RegisterExtension() proto.pokerth_pb2.InitMessage static method), 57	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListAdminChangedMessage static method), 41	(pokerth-	RegisterExtension() proto.pokerth_pb2.InviteNotifyMessage method), 58	static
RegisterExtension() proto.pokerth_pb2.GameListNewMessage static method), 42	(pokerth-	RegisterExtension() proto.pokerth_pb2.InvitePlayerToGameMessage static method), 59	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListPlayerJoinedMessage static method), 43	(pokerth-	RegisterExtension() proto.pokerth_pb2.JoinExistingGameMessage static method), 60	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListPlayerLeftMessage static method), 44	(pokerth-	RegisterExtension() proto.pokerth_pb2.JoinGameAckMessage static method), 61	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListSpectatorJoinedMessage static method), 45	(pokerth-	RegisterExtension() proto.pokerth_pb2.JoinGameFailedMessage static method), 62	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListSpectatorLeftMessage static method), 45	(pokerth-	RegisterExtension() proto.pokerth_pb2.JoinNewGameMessage static method), 63	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameListUpdateMessage static method), 46	(pokerth-	RegisterExtension() proto.pokerth_pb2.KickPetitionUpdateMessage static method), 64	(pokerth-
RegisterExtension() proto.pokerth_pb2.GamePlayerJoinedMessage static method), 47	(pokerth-	RegisterExtension() proto.pokerth_pb2.KickPlayerRequestMessage static method), 65	(pokerth-
RegisterExtension() proto.pokerth_pb2.GamePlayerLeftMessage static method), 48	(pokerth-	RegisterExtension() proto.pokerth_pb2.LeaveGameRequestMessage static method), 66	(pokerth-
RegisterExtension() proto.pokerth_pb2.GameSpectatorJoinedMessage static method), 49	(pokerth-	RegisterExtension() proto.pokerth_pb2.MyActionRequestMessage static method), 67	(pokerth-

RegisterExtension() proto.pokerth_pb2.NetGameInfo static method), 68	(pokerth- static	RegisterExtension() proto.pokerth_pb2.ReportGameMessage static method), 95	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerIdChangedMessage static method), 70	(pokerth-	RegisterExtension() proto.pokerth_pb2.ResetTimeoutMessage static method), 96	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerInfoReplyMessage static method), 73	(pokerth-	RegisterExtension() proto.pokerth_pb2.ShowMyCardsRequestMessage static method), 96	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData static method), 73	(pokerth-	RegisterExtension() proto.pokerth_pb2.StartEventAckMessage static method), 97	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoDataAvatarData static method), 72	(pokerth-	RegisterExtension() proto.pokerth_pb2.StartEventMessage static method), 98	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerInfoRequestMessage static method), 74	(pokerth-	RegisterExtension() proto.pokerth_pb2.StartKickPetitionMessage static method), 99	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayerListMessage static method), 75	(pokerth- static	RegisterExtension() proto.pokerth_pb2.StatisticsMessage static method), 100	(pokerth- static
RegisterExtension() proto.pokerth_pb2.PlayerResult static method), 76	(pokerth- static	RegisterExtension() proto.pokerth_pb2.StatisticsMessage.StatisticsData static method), 101	(pokerth- static
RegisterExtension() proto.pokerth_pb2.PlayersActionDoneMessage static method), 77	(pokerth-	RegisterExtension() proto.pokerth_pb2.SubscriptionRequestMessage static method), 102	(pokerth-
RegisterExtension() proto.pokerth_pb2.PlayersTurnMessage static method), 78	(pokerth- static	RegisterExtension() proto.pokerth_pb2.TimeoutWarningMessage static method), 103	(pokerth- static
RegisterExtension() proto.pokerth_pb2.PokerTHMessage static method), 81	(pokerth- static	RegisterExtension() proto.pokerth_pb2.UnknownAvatarMessage static method), 104	(pokerth- static
RegisterExtension() proto.pokerth_pb2.RejectGameInvitationMessage static method), 88	(pokerth-	RegisterExtension() proto.pokerth_pb2.VoteKickReplyMessage static method), 104	(pokerth-
RegisterExtension() proto.pokerth_pb2.RejectInvNotifyMessage static method), 89	(pokerth-	RegisterExtension() proto.pokerth_pb2.VoteKickRequestMessage static method), 106	(pokerth-
RegisterExtension() proto.pokerth_pb2.RejoinExistingGameMessage static method), 90	(pokerth-	RegisterExtension() proto.pokerth_pb2>YourActionRejectedMessage static method), 107	(pokerth-
RegisterExtension() proto.pokerth_pb2.RemovedFromGameMessage static method), 91	(pokerth-	registerServer() pokerthproto.proxy.ProxyProtocol method), 112	(pokerth-
RegisterExtension() proto.pokerth_pb2.ReportAvatarAckMessage static method), 92	(pokerth-	rejectedActionNotAllowed proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth-
RegisterExtension() proto.pokerth_pb2.ReportAvatarMessage static method), 93	(pokerth-	rejectedInvalidGameState proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth-
RegisterExtension() proto.pokerth_pb2.ReportGameAckMessage static method), 94	(pokerth-	rejectedNotYourTurn proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth-

RejectGameInvitationMessage (class in pokerthproto.pokerth_pb2), 88	rejoinFailed (pokerthproto.pokerth_pb2.JoinGameFailedMessage attribute), 63
rejectGameInvitationMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87	rejoinGameId (pokerthproto.pokerth_pb2.InitAckMessage attribute), 56
REJECTGAMEINVITATIONMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	REJOIN_GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitAckMessage attribute), 56
rejectGameInvitationReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	rejoinPlayerData (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53
RejectGameInvReason (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 88	REJOINPLAYERDATA_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 52
RejectInvNotifyMessage (class in pokerthproto.pokerth_pb2), 89	remainingSeconds (pokerthproto.pokerth_pb2.TimeoutWarningMessage attribute), 103
rejectInvNotifyMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87	REMAININGSECONDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.TimeoutWarningMessage attribute), 103
REJECTINVNOTIFYMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	RemovedFromGameMessage (class in pokerthproto.pokerth_pb2), 90
rejectInvNotifyReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	removedFromGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
RejectionReason (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 107	REMOVEDFROMGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
rejectionReason (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 107	RemovedFromGameReason (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
REJECTIONREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 106	removedFromGameReason (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
rejectReasonBusy (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 89	REMOVEDFROMGAMEREASON_FIELD_NUMBER (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
rejectReasonNo (pokerthproto.pokerth_pb2.RejectGameInvitationMessage attribute), 89	removedFromGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
rejoinEvent (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98	removedOnRequest (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
RejoinExistingGameMessage (class in pokerthproto.pokerth_pb2), 90	removedStartFailed (pokerthproto.pokerth_pb2.RemovedFromGameMessage attribute), 91
rejoinExistingGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87	removeGameId (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14
REJOINEXISTINGGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81	removeGameId (pokerthproto.pokerth_pb2.AdminRemoveGameMessage attribute), 15
rejoinExistingGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111	REMOVE_GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14
	REMOVE_GAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 15
	REMOVEGAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 15

attribute), 14

REMOVEGAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AdminRemoveGameMessage attribute), 15

removeGameResult (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14

REMOVEGAMERESULT_FIELD_NUMBER (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage attribute), 14

ReportAvatarAckMessage (class in pokerthproto.pokerth_pb2), 92

reportAvatarAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87

REPORTAVATARACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81

reportAvatarAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 111

ReportAvatarMessage (class in pokerthproto.pokerth_pb2), 92

reportAvatarMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87

REPORTAVATARMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81

reportAvatarReceived() (pokerthproto.protocol.PokerTHProtocol method), 111

ReportAvatarResult (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92

reportAvatarResult (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92

REPORTAVATARRESULT_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92

reportedAvatarHash (pokerthproto.pokerth_pb2.ReportAvatarMessage attribute), 93

REPORTEDAVATARHASH_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportAvatarMessage attribute), 93

reportedGameId (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94

reportedGameId (pokerthproto.pokerth_pb2.ReportGameMessage attribute), 95

REPORTEDGAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94

REPORTEDGAMEID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportGameMessage attribute), 95

reportedPlayerId (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92

reportedPlayerId (pokerthproto.pokerth_pb2.ReportAvatarMessage attribute), 93

REPORTEDPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportAvatarAckMessage attribute), 92

REPORTEDPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportAvatarMessage attribute), 93

ReportGameAckMessage (class in pokerthproto.pokerth_pb2), 93

reportGameAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87

REPORTGAMEACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81

reportGameAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 111

ReportGameMessage (class in pokerthproto.pokerth_pb2), 94

reportGameMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87

REPORTGAMEMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81

reportGameReceived() (pokerthproto.protocol.PokerTHProtocol method), 111

ReportGameResult (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94

reportGameResult (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94

REPORTGAMERESULT_FIELD_NUMBER (pokerthproto.pokerth_pb2.ReportGameAckMessage attribute), 94

requestedVersion (pokerthproto.pokerth_pb2.InitMessage attribute), 58

REQUESTEDVERSION_FIELD_NUMBER (pokerthproto.pokerth_pb2.InitMessage attribute),

- 57
- requestId (pokerthproto.pokerth_pb2.AvatarDataMessage attribute), 25
- requestId (pokerthproto.pokerth_pb2.AvatarEndMessage attribute), 26
- requestId (pokerthproto.pokerth_pb2.AvatarHeaderMessage attribute), 27
- requestId (pokerthproto.pokerth_pb2.AvatarRequestMessage attribute), 28
- requestId (pokerthproto.pokerth_pb2.UnknownAvatarMessage attribute), 104
- REQUESTID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarDataMessage attribute), 25
- REQUESTID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarEndMessage attribute), 26
- REQUESTID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarHeaderMessage attribute), 26
- REQUESTID_FIELD_NUMBER (pokerthproto.pokerth_pb2.AvatarRequestMessage attribute), 27
- REQUESTID_FIELD_NUMBER (pokerthproto.pokerth_pb2.UnknownAvatarMessage attribute), 104
- reserved (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39
- ResetTimeoutMessage (class in pokerthproto.pokerth_pb2), 95
- resetTimeoutMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
- RESETTIMEOUTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
- resetTimeoutReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
- resubscribeGameList (pokerthproto.pokerth_pb2.SubscriptionRequestMessage attribute), 102
- resultCard1 (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
- RESULTCARD1_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
- resultCard2 (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
- RESULTCARD2_FIELD_NUMBER (pokerthproto.pokerth_pb2.PlayerResult attribute), 76
- resultPlayerKicked (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- RESULTPLAYERKICKED_FIELD_NUMBER (pokerthproto.pokerth_pb2.EndKickPetitionMessage attribute), 35
- RIVER (pokerthproto.poker.Round attribute), 11
- riverCard (pokerthproto.pokerth_pb2.DealRiverCardMessage attribute), 32
- RIVERCARD_FIELD_NUMBER (pokerthproto.pokerth_pb2.DealRiverCardMessage attribute), 32
- Round (class in pokerthproto.poker), 11
- RoundInfo (class in pokerthproto.game), 9
- ## S
- seat (pokerthproto.player.Player attribute), 11
- seats (pokerthproto.game.Game attribute), 9
- seatStates (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
- SEATSTATES_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
- sendChatRequest() (pokerthproto.protocol.ClientProtocol method), 109
- sendJoinExistingGame() (pokerthproto.protocol.ClientProtocol method), 109
- sendJoinNewGame() (pokerthproto.protocol.ClientProtocol method), 109
- sendMyAction() (pokerthproto.protocol.ClientProtocol method), 109
- sendStartEvent() (pokerthproto.protocol.ClientProtocol method), 109
- sendToClient() (pokerthproto.proxy.ProxyProtocol method), 112
- SerializePartialToString() (pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12
- SerializePartialToString() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13
- SerializePartialToString() (pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14
- SerializePartialToString() (pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15
- SerializePartialToString() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 16
- SerializePartialToString() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 17
- SerializePartialToString() (pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17

SerializePartialToString() proto.pokerth_pb2.AnnounceMessage method), 18	(pokerth-	SerializePartialToString() proto.pokerth_pb2.EndKickPetitionMessage method), 35	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AnnounceMessage.Version method), 19	(pokerth-	SerializePartialToString() proto.pokerth_pb2.EndOfGameMessage method), 36	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AskKickDeniedMessage method), 20	(pokerth-	SerializePartialToString() proto.pokerth_pb2.EndOfHandHideCardsMessage method), 37	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AskKickPlayerMessage method), 21	(pokerth-	SerializePartialToString() proto.pokerth_pb2.EndOfHandShowCardsMessage method), 38	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AuthClientResponseMessage method), 22	(pokerth-	SerializePartialToString() proto.pokerth_pb2.ErrorMessage 39	(pokerth- method),
SerializePartialToString() proto.pokerth_pb2.AuthServerChallengeMessage method), 23	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameAdminChangedMessage method), 40	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AuthServerVerificationMessage method), 24	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListAdminChangedMessage method), 41	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AvatarDataMessage method), 25	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListNewMessage method), 42	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AvatarEndMessage method), 26	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AvatarHeaderMessage method), 26	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListPlayerLeftMessage method), 44	(pokerth-
SerializePartialToString() proto.pokerth_pb2.AvatarRequestMessage method), 27	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45	(pokerth-
SerializePartialToString() proto.pokerth_pb2.ChatMessage 28	(pokerth- method),	SerializePartialToString() proto.pokerth_pb2.GameListSpectatorLeftMessage method), 45	(pokerth-
SerializePartialToString() proto.pokerth_pb2.ChatRejectMessage method), 29	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameListUpdateMessage method), 46	(pokerth-
SerializePartialToString() proto.pokerth_pb2.ChatRequestMessage method), 30	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GamePlayerJoinedMessage method), 47	(pokerth-
SerializePartialToString() proto.pokerth_pb2.DealFlopCardsMessage method), 31	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GamePlayerLeftMessage method), 48	(pokerth-
SerializePartialToString() proto.pokerth_pb2.DealRiverCardMessage method), 32	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	(pokerth-
SerializePartialToString() proto.pokerth_pb2.DealTurnCardMessage method), 33	(pokerth-	SerializePartialToString() proto.pokerth_pb2.GameSpectatorLeftMessage method), 50	(pokerth-
SerializePartialToString() proto.pokerth_pb2.DialogMessage 34	(pokerth- method),	SerializePartialToString() proto.pokerth_pb2.GameStartInitialMessage method), 51	(pokerth-

SerializePartialToString() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 53	SerializePartialToString() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 53	SerializePartialToString() (pokerthproto.pokerth_pb2.HandStartMessage method), 55	SerializePartialToString() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 54	SerializePartialToString() (pokerthproto.pokerth_pb2.InitAckMessage method), 56	SerializePartialToString() (pokerthproto.pokerth_pb2.InitMessage method), 57	SerializePartialToString() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58	SerializePartialToString() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59	SerializePartialToString() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60	SerializePartialToString() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61	SerializePartialToString() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62	SerializePartialToString() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63	SerializePartialToString() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64	SerializePartialToString() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65	SerializePartialToString() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66	SerializePartialToString() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67	SerializePartialToString() (pokerthproto.pokerth_pb2.NetGameInfo method), 69	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 73	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 73	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.Avatar method), 72	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerListMessage method), 75	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayerResult method), 76	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77	SerializePartialToString() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78	SerializePartialToString() (pokerthproto.pokerth_pb2.PokerTHMessage method), 81	SerializePartialToString() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88	SerializePartialToString() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89	SerializePartialToString() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90	SerializePartialToString() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91	SerializePartialToString() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92	SerializePartialToString() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93	SerializePartialToString() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94	SerializePartialToString() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95	SerializePartialToString() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 96
--	---	--	---	--	---	---	---	---	--	---	--	---	--	---	--	---	--	--	---	--	--	---	--	--	--	--	---	--	---	--	--	---	--	---	---

SerializePartialToString() proto.pokerth_pb2.ShowMyCardsRequestMessage method), 96	(pokerth-	SerializeToString() proto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17	(pokerth-
SerializePartialToString() proto.pokerth_pb2.StartEventAckMessage method), 97	(pokerth-	SerializeToString() proto.pokerth_pb2.AnnounceMessage method), 18	(pokerth-
SerializePartialToString() proto.pokerth_pb2.StartEventMessage method), 98	(pokerth-	SerializeToString() proto.pokerth_pb2.AnnounceMessage.Version method), 19	(pokerth-
SerializePartialToString() proto.pokerth_pb2.StartKickPetitionMessage method), 99	(pokerth-	SerializeToString() proto.pokerth_pb2.AskKickDeniedMessage method), 20	(pokerth-
SerializePartialToString() proto.pokerth_pb2.StatisticsMessage method), 100	(pokerth-	SerializeToString() proto.pokerth_pb2.AskKickPlayerMessage method), 22	(pokerth-
SerializePartialToString() proto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101	(pokerth-	SerializeToString() proto.pokerth_pb2.AuthClientResponseMessage method), 22	(pokerth-
SerializePartialToString() proto.pokerth_pb2.SubscriptionRequestMessage method), 102	(pokerth-	SerializeToString() proto.pokerth_pb2.AuthServerChallengeMessage method), 23	(pokerth-
SerializePartialToString() proto.pokerth_pb2.TimeoutWarningMessage method), 103	(pokerth-	SerializeToString() proto.pokerth_pb2.AuthServerVerificationMessage method), 24	(pokerth-
SerializePartialToString() proto.pokerth_pb2.UnknownAvatarMessage method), 104	(pokerth-	SerializeToString() proto.pokerth_pb2.AvatarDataMessage method), 25	(pokerth-
SerializePartialToString() proto.pokerth_pb2.VoteKickReplyMessage method), 105	(pokerth-	SerializeToString() proto.pokerth_pb2.AvatarEndMessage method), 26	(pokerth-
SerializePartialToString() proto.pokerth_pb2.VoteKickRequestMessage method), 106	(pokerth-	SerializeToString() proto.pokerth_pb2.AvatarHeaderMessage method), 26	(pokerth-
SerializePartialToString() proto.pokerth_pb2>YourActionRejectedMessage method), 107	(pokerth-	SerializeToString() proto.pokerth_pb2.AvatarRequestMessage method), 27	(pokerth-
SerializeToString() proto.pokerth_pb2.AdminBanPlayerAckMessage method), 12	(pokerth-	SerializeToString() proto.pokerth_pb2.ChatMessage method), 28	(pokerth-
SerializeToString() proto.pokerth_pb2.AdminBanPlayerMessage method), 13	(pokerth-	SerializeToString() proto.pokerth_pb2.ChatRejectMessage method), 29	(pokerth-
SerializeToString() proto.pokerth_pb2.AdminRemoveGameAckMessage method), 14	(pokerth-	SerializeToString() proto.pokerth_pb2.ChatRequestMessage method), 30	(pokerth-
SerializeToString() proto.pokerth_pb2.AdminRemoveGameMessage method), 15	(pokerth-	SerializeToString() proto.pokerth_pb2.DealFlopCardsMessage method), 31	(pokerth-
SerializeToString() proto.pokerth_pb2.AfterHandShowCardsMessage method), 16	(pokerth-	SerializeToString() proto.pokerth_pb2.DealRiverCardMessage method), 32	(pokerth-
SerializeToString() proto.pokerth_pb2.AllInShowCardsMessage method), 17	(pokerth-	SerializeToString() proto.pokerth_pb2.DealTurnCardMessage method), 33	(pokerth-

SerializeToString() proto.pokerth_pb2.DialogMessage method), 34	(pokerth-	SerializeToString() proto.pokerth_pb2.GameStartInitialMessage method), 51	(pokerth-
SerializeToString() proto.pokerth_pb2.EndKickPetitionMessage method), 35	(pokerth-	SerializeToString() proto.pokerth_pb2.GameStartRejoinMessage method), 53	(pokerth-
SerializeToString() proto.pokerth_pb2.EndOfGameMessage method), 36	(pokerth-	SerializeToString() proto.pokerth_pb2.GameStartRejoinMessage.RejoinPlayerData method), 53	(pokerth-
SerializeToString() proto.pokerth_pb2.EndOfHandHideCardsMessage method), 37	(pokerth-	SerializeToString() proto.pokerth_pb2.HandStartMessage method), 55	(pokerth-
SerializeToString() proto.pokerth_pb2.EndOfHandShowCardsMessage method), 38	(pokerth-	SerializeToString() proto.pokerth_pb2.HandStartMessage.PlainCards method), 54	(pokerth-
SerializeToString() proto.pokerth_pb2.ErrorMessage method), 39	(pokerth-	SerializeToString() proto.pokerth_pb2.InitAckMessage method), 56	(pokerth-
SerializeToString() proto.pokerth_pb2.GameAdminChangedMessage method), 40	(pokerth-	SerializeToString() proto.pokerth_pb2.InitMessage method), 57	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListAdminChangedMessage method), 41	(pokerth-	SerializeToString() proto.pokerth_pb2.InviteNotifyMessage method), 58	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListNewMessage method), 42	(pokerth-	SerializeToString() proto.pokerth_pb2.InvitePlayerToGameMessage method), 59	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	(pokerth-	SerializeToString() proto.pokerth_pb2.JoinExistingGameMessage method), 60	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListPlayerLeftMessage method), 44	(pokerth-	SerializeToString() proto.pokerth_pb2.JoinGameAckMessage method), 61	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45	(pokerth-	SerializeToString() proto.pokerth_pb2.JoinGameFailedMessage method), 62	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListSpectatorLeftMessage method), 46	(pokerth-	SerializeToString() proto.pokerth_pb2.JoinNewGameMessage method), 63	(pokerth-
SerializeToString() proto.pokerth_pb2.GameListUpdateMessage method), 46	(pokerth-	SerializeToString() proto.pokerth_pb2.KickPetitionUpdateMessage method), 64	(pokerth-
SerializeToString() proto.pokerth_pb2.GamePlayerJoinedMessage method), 47	(pokerth-	SerializeToString() proto.pokerth_pb2.KickPlayerRequestMessage method), 65	(pokerth-
SerializeToString() proto.pokerth_pb2.GamePlayerLeftMessage method), 48	(pokerth-	SerializeToString() proto.pokerth_pb2.LeaveGameRequestMessage method), 66	(pokerth-
SerializeToString() proto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	(pokerth-	SerializeToString() proto.pokerth_pb2.MyActionRequestMessage method), 67	(pokerth-
SerializeToString() proto.pokerth_pb2.GameSpectatorLeftMessage method), 50	(pokerth-	SerializeToString() proto.pokerth_pb2.NetGameInfo method), 69	(pokerth-

SerializeToString() proto.pokerth_pb2.PlayerIdChangedMessage method), 70	(pokerth-	SerializeToString() proto.pokerth_pb2.ResetTimeoutMessage method), 96	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerInfoReplyMessage method), 73	(pokerth-	SerializeToString() proto.pokerth_pb2.ShowMyCardsRequestMessage method), 96	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 73	(pokerth-	SerializeToString() proto.pokerth_pb2.StartEventAckMessage method), 97	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoDataAvatarData method), 72	(pokerth-	SerializeToString() proto.pokerth_pb2.StartEventMessage method), 98	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerInfoRequestMessage method), 74	(pokerth-	SerializeToString() proto.pokerth_pb2.StartKickPetitionMessage method), 99	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerListMessage method), 75	(pokerth-	SerializeToString() proto.pokerth_pb2.StatisticsMessage method), 100	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayerResult method), 76	(pokerth-	SerializeToString() proto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayersActionDoneMessage method), 77	(pokerth-	SerializeToString() proto.pokerth_pb2.SubscriptionRequestMessage method), 102	(pokerth-
SerializeToString() proto.pokerth_pb2.PlayersTurnMessage method), 78	(pokerth-	SerializeToString() proto.pokerth_pb2.TimeoutWarningMessage method), 103	(pokerth-
SerializeToString() proto.pokerth_pb2.PokerTHMessage method), 81	(pokerth-	SerializeToString() proto.pokerth_pb2.UnknownAvatarMessage method), 104	(pokerth-
SerializeToString() proto.pokerth_pb2.RejectGameInvitationMessage method), 88	(pokerth-	SerializeToString() proto.pokerth_pb2.VoteKickReplyMessage method), 105	(pokerth-
SerializeToString() proto.pokerth_pb2.RejectInvNotifyMessage method), 89	(pokerth-	SerializeToString() proto.pokerth_pb2.VoteKickRequestMessage method), 106	(pokerth-
SerializeToString() proto.pokerth_pb2.RejoinExistingGameMessage method), 90	(pokerth-	SerializeToString() proto.pokerth_pb2.YourActionRejectedMessage method), 107	(pokerth-
SerializeToString() proto.pokerth_pb2.RemovedFromGameMessage method), 91	(pokerth-	serverChallenge proto.pokerth_pb2.AuthServerChallengeMessage attribute), 23	(pokerth-
SerializeToString() proto.pokerth_pb2.ReportAvatarAckMessage method), 92	(pokerth-	SERVERCHALLENGE_FIELD_NUMBER proto.pokerth_pb2.AuthServerChallengeMessage attribute), 23	(pokerth-
SerializeToString() proto.pokerth_pb2.ReportAvatarMessage method), 93	(pokerth-	ServerType (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 18	
SerializeToString() proto.pokerth_pb2.ReportGameAckMessage method), 94	(pokerth-	serverType (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 20	
SerializeToString() proto.pokerth_pb2.ReportGameMessage method), 95	(pokerth-	SERVERTYPE_FIELD_NUMBER proto.pokerth_pb2.AnnounceMessage attribute), 18	(pokerth-
		serverTypeInternetAuth proto.pokerth_pb2.AnnounceMessage attribute), 18	(pokerth-

tribute), 20		SetInParent() (pokerthproto.pokerth_pb2.ChatMessage method), 28
serverTypeInternetNoAuth (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 20		SetInParent() (pokerthproto.pokerth_pb2.ChatRejectMessage method), 29
serverTypeLAN (pokerthproto.pokerth_pb2.AnnounceMessage attribute), 20		SetInParent() (pokerthproto.pokerth_pb2.ChatRequestMessage method), 30
serverVerification (pokerthproto.pokerth_pb2.AuthServerVerificationMessage attribute), 24		SetInParent() (pokerthproto.pokerth_pb2.DealFlopCardsMessage method), 31
SERVERVERIFICATION_FIELD_NUMBER (pokerthproto.pokerth_pb2.AuthServerVerificationMessage attribute), 24		SetInParent() (pokerthproto.pokerth_pb2.DealRiverCardMessage method), 32
sessionTimeout (pokerthproto.pokerth_pb2.ErrorMessage attribute), 39		SetInParent() (pokerthproto.pokerth_pb2.DealTurnCardMessage method), 33
setInfo() (pokerthproto.lobby.GameInfo method), 10		SetInParent() (pokerthproto.pokerth_pb2.DialogMessage method), 34
setInfo() (pokerthproto.player.Player method), 11		SetInParent() (pokerthproto.pokerth_pb2.EndKickPetitionMessage method), 35
SetInParent() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 12		SetInParent() (pokerthproto.pokerth_pb2.EndOfGameMessage method), 36
SetInParent() (pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13		SetInParent() (pokerthproto.pokerth_pb2.EndOfHandHideCardsMessage method), 37
SetInParent() (pokerthproto.pokerth_pb2.AdminRemoveGameAdminMessage method), 14		SetInParent() (pokerthproto.pokerth_pb2.EndOfHandShowCardsMessage method), 38
SetInParent() (pokerthproto.pokerth_pb2.AdminRemoveGameAdminMessage method), 15		SetInParent() (pokerthproto.pokerth_pb2.ErrorMessage method), 39
SetInParent() (pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 16		SetInParent() (pokerthproto.pokerth_pb2.GameAdminChangedMessage method), 40
SetInParent() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 18		SetInParent() (pokerthproto.pokerth_pb2.GameListAdminChangedMessage method), 41
SetInParent() (pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 17		SetInParent() (pokerthproto.pokerth_pb2.GameListNewMessage method), 42
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 18		SetInParent() (pokerthproto.pokerth_pb2.GameListPlayerJoinedMessage method), 43
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 19		SetInParent() (pokerthproto.pokerth_pb2.GameListPlayerLeftMessage method), 44
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 20		SetInParent() (pokerthproto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 21		SetInParent() (pokerthproto.pokerth_pb2.GameListSpectatorLeftMessage method), 46
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 22		SetInParent() (pokerthproto.pokerth_pb2.GameListUpdateMessage method), 46
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 23		SetInParent() (pokerthproto.pokerth_pb2.GamePlayerJoinedMessage method), 47
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 24		SetInParent() (pokerthproto.pokerth_pb2.GamePlayerLeftMessage method), 48
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 25		SetInParent() (pokerthproto.pokerth_pb2.GameSpectatorJoinedMessage method), 49
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 26		SetInParent() (pokerthproto.pokerth_pb2.GameSpectatorLeftMessage method), 50
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 27		SetInParent() (pokerthproto.pokerth_pb2.GameStartInitialMessage method), 51
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 28		SetInParent() (pokerthproto.pokerth_pb2.GameStartRejoinMessage method), 53
SetInParent() (pokerthproto.pokerth_pb2.AnnounceMessage method), 29		SetInParent() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.Rejoin method), 53

SetInParent() (pokerthproto.pokerth_pb2.HandStartMessageSetInParent() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 55 method), 90

SetInParent() (pokerthproto.pokerth_pb2.HandStartMessageSetInParent() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 54 method), 91

SetInParent() (pokerthproto.pokerth_pb2.InitAckMessage SetInParent() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 56 method), 92

SetInParent() (pokerthproto.pokerth_pb2.InitMessage SetInParent() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 57 method), 93

SetInParent() (pokerthproto.pokerth_pb2.InviteNotifyMessageSetInParent() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 58 method), 94

SetInParent() (pokerthproto.pokerth_pb2.InvitePlayerToGameSetInParent() (pokerthproto.pokerth_pb2.ReportGameMessage method), 59 method), 95

SetInParent() (pokerthproto.pokerth_pb2.JoinExistingGameMessageSetInParent() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 60 method), 96

SetInParent() (pokerthproto.pokerth_pb2.JoinGameAckMessageSetInParent() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 61 method), 96

SetInParent() (pokerthproto.pokerth_pb2.JoinGameFailedMessageSetInParent() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 62 method), 97

SetInParent() (pokerthproto.pokerth_pb2.JoinNewGameMessageSetInParent() (pokerthproto.pokerth_pb2.StartEventMessage method), 63 method), 98

SetInParent() (pokerthproto.pokerth_pb2.KickPetitionUpdateSMISetInParent() (pokerthproto.pokerth_pb2.StartKickPetitionMessage method), 64 method), 99

SetInParent() (pokerthproto.pokerth_pb2.KickPlayerRequestSMISetInParent() (pokerthproto.pokerth_pb2.StatisticsMessage method), 65 method), 100

SetInParent() (pokerthproto.pokerth_pb2.LeaveGameRequestSMISetInParent() (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData method), 66 method), 101

SetInParent() (pokerthproto.pokerth_pb2.MyActionRequestSMISetInParent() (pokerthproto.pokerth_pb2.SubscriptionRequestMessage method), 67 method), 102

SetInParent() (pokerthproto.pokerth_pb2.NetGameInfo SetInParent() (pokerthproto.pokerth_pb2.TimeoutWarningMessage method), 69 method), 103

SetInParent() (pokerthproto.pokerth_pb2.PlayerIdChangedMessageSetInParent() (pokerthproto.pokerth_pb2.UnknownAvatarMessage method), 70 method), 104

SetInParent() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessageSetInParent() (pokerthproto.pokerth_pb2.VoteKickReplyMessage method), 73 method), 105

SetInParent() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessageSetInParent() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 73 method), 106

SetInParent() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessageSetInParent() (pokerthproto.pokerth_pb2.VoteKickRequestMessage method), 72 method), 107

SetInParent() (pokerthproto.pokerth_pb2.PlayerInfoRequestSMISetInParent() (pokerthproto.pokerth_pb2>YourActionRejectedMessage method), 74 method), 107

SetInParent() (pokerthproto.pokerth_pb2.PlayerListMessage ShowMyCardsRequestMessage (class in pokerthproto.pokerth_pb2), 96

SetInParent() (pokerthproto.pokerth_pb2.PlayerResult showMyCardsRequestMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87

SetInParent() (pokerthproto.pokerth_pb2.PlayersActionDoneSMISetInParent() (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 77 method), 81

SetInParent() (pokerthproto.pokerth_pb2.PlayersTurnMessage showMyCardsRequestReceived() (pokerthproto.protocol.ClientProtocol method), 109

SetInParent() (pokerthproto.pokerth_pb2.PokerTHMessage showMyCardsRequestReceived() (pokerthproto.protocol.PokerTHProtocol method), 81

SetInParent() (pokerthproto.pokerth_pb2.RejectGameInvitationMessageSMISetInParent() (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 88 method), 111

SetInParent() (pokerthproto.pokerth_pb2.RejectInvNotifyMessageSMISetInParent() (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 89 method), 111

smallBlind (pokerthproto.game.Game attribute), 9
 smallBlind (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
 SMALLBLIND_FIELD_NUMBER (pokerthproto.pokerth_pb2.HandStartMessage attribute), 55
 spectateOnly (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60
 spectateOnly (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61
 SPECTATEONLY_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinExistingGameMessage attribute), 60
 SPECTATEONLY_FIELD_NUMBER (pokerthproto.pokerth_pb2.JoinGameAckMessage attribute), 61
 spectatorIds (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42
 SPECTATORIDS_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameListNewMessage attribute), 42
 startDealerPlayerId (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51
 startDealerPlayerId (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53
 STARTDEALERPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartInitialMessage attribute), 51
 STARTDEALERPLAYERID_FIELD_NUMBER (pokerthproto.pokerth_pb2.GameStartRejoinMessage attribute), 53
 startEvent (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98
 StartEventAckMessage (class in pokerthproto.pokerth_pb2), 97
 startEventAckMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
 STARTEVENTACKMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
 startEventAckReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 StartEventMessage (class in pokerthproto.pokerth_pb2), 97
 startEventMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
 STARTEVENTMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
 startEventReceived() (pokerthproto.protocol.ClientProtocol method), 109
 startEventReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 StartEventType (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98
 StartEventType (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98
 STARTEVENTTYPE_FIELD_NUMBER (pokerthproto.pokerth_pb2.StartEventMessage attribute), 98
 StartKickPetitionMessage (class in pokerthproto.pokerth_pb2), 98
 startKickPetitionMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
 STARTKICKPETITIONMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
 startKickPetitionReceived() (pokerthproto.protocol.PokerTHProtocol method), 111
 startMoney (pokerthproto.lobby.GameInfo attribute), 10
 startMoney (pokerthproto.pokerth_pb2.NetGameInfo attribute), 70
 STARTMONEY_FIELD_NUMBER (pokerthproto.pokerth_pb2.NetGameInfo attribute), 68
 startNewHand() (pokerthproto.game.Game method), 9
 state (pokerthproto.protocol.ClientProtocol attribute), 109
 States (class in pokerthproto.protocol), 112
 statisticsData (pokerthproto.pokerth_pb2.StatisticsMessage attribute), 101
 STATISTICSDATA_FIELD_NUMBER (pokerthproto.pokerth_pb2.StatisticsMessage attribute), 100
 StatisticsMessage (class in pokerthproto.pokerth_pb2), 100
 statisticsMessage (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87
 StatisticsMessage.StatisticsData (class in pokerthproto.pokerth_pb2), 100
 STATISTICSMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 81
 statisticsReceived() (pokerthproto.protocol.PokerTHProtocol method), 112
 StatisticsType (pokerthproto.pokerth_pb2.StatisticsMessage.StatisticsData

attribute), 101	timeoutKickAfterAutofold (pokerth-
statisticsType (pokerth-	proto.pokerth_pb2.TimeoutWarningMessage
proto.pokerth_pb2.StatisticsMessage.StatisticsData	attribute), 103
attribute), 101	timeoutNoDataReceived (pokerth-
STATISTICSTYPE_FIELD_NUMBER (pokerth-	proto.pokerth_pb2.TimeoutWarningMessage
proto.pokerth_pb2.StatisticsMessage.StatisticsData	attribute), 103
attribute), 101	TimeoutReason (pokerth-
statisticsValue (pokerth-	proto.pokerth_pb2.TimeoutWarningMessage
proto.pokerth_pb2.StatisticsMessage.StatisticsData	attribute), 103
attribute), 101	timeoutReason (pokerth-
STATISTICSVALUE_FIELD_NUMBER (pokerth-	proto.pokerth_pb2.TimeoutWarningMessage
proto.pokerth_pb2.StatisticsMessage.StatisticsData	attribute), 103
attribute), 101	TIMEOUTREASON_FIELD_NUMBER (pokerth-
statNumberOfPlayers (pokerth-	proto.pokerth_pb2.TimeoutWarningMessage
proto.pokerth_pb2.StatisticsMessage.StatisticsData	attribute), 103
attribute), 101	TimeoutWarningMessage (class in pokerth-
SubscriptionAction (pokerth-	proto.pokerth_pb2), 102
proto.pokerth_pb2.SubscriptionRequestMessage	timeoutWarningMessage (pokerth-
attribute), 102	proto.pokerth_pb2.PokerTHMessage attribute),
subscriptionAction (pokerth-	87
proto.pokerth_pb2.SubscriptionRequestMessage	TIMEOUTWARNINGMESSAGE_FIELD_NUMBER
attribute), 102	(pokerthproto.pokerth_pb2.PokerTHMessage
SUBSCRIPTIONACTION_FIELD_NUMBER (pokerth-	attribute), 81
proto.pokerth_pb2.SubscriptionRequestMessage	timeoutWarningReceived() (pokerth-
attribute), 102	proto.protocol.PokerTHProtocol method),
SubscriptionRequestMessage (class in pokerth-	112
proto.pokerth_pb2), 101	totalPlayerBet (pokerth-
subscriptionRequestMessage (pokerth-	proto.pokerth_pb2.PlayersActionDoneMessage
proto.pokerth_pb2.PokerTHMessage attribute),	attribute), 77
87	TOTALPLAYERBET_FIELD_NUMBER (pokerth-
SUBSCRIPTIONREQUESTMESSAGE_FIELD_NUMBER	proto.pokerth_pb2.PlayersActionDoneMessage
(pokerthproto.pokerth_pb2.PokerTHMessage	attribute), 77
attribute), 81	TURN (pokerthproto.poker.Round attribute), 11
subscriptionRequestReceived() (pokerth-	turnCard (pokerthproto.pokerth_pb2.DealTurnCardMessage
proto.protocol.PokerTHProtocol method),	attribute), 33
112	TURNCARD_FIELD_NUMBER (pokerth-
T	proto.pokerth_pb2.DealTurnCardMessage
targetGameId (pokerth-	attribute), 33
proto.pokerth_pb2.ChatRequestMessage	Type_AdminBanPlayerAckMessage (pokerth-
attribute), 30	proto.pokerth_pb2.PokerTHMessage attribute),
TARGETGAMEID_FIELD_NUMBER (pokerth-	81
proto.pokerth_pb2.ChatRequestMessage	Type_AdminBanPlayerMessage (pokerth-
attribute), 30	proto.pokerth_pb2.PokerTHMessage attribute),
targetPlayerId (pokerth-	81
proto.pokerth_pb2.ChatRequestMessage	Type_AdminRemoveGameAckMessage (pokerth-
attribute), 30	proto.pokerth_pb2.PokerTHMessage attribute),
TARGETPLAYERID_FIELD_NUMBER (pokerth-	81
proto.pokerth_pb2.ChatRequestMessage	Type_AdminRemoveGameMessage (pokerth-
attribute), 30	proto.pokerth_pb2.PokerTHMessage attribute),
timeoutInactiveGame (pokerth-	81
proto.pokerth_pb2.TimeoutWarningMessage	Type_AfterHandShowCardsMessage (pokerth-
attribute), 103	proto.pokerth_pb2.PokerTHMessage attribute),
	81

Type_AllInShowCardsMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_EndKickPetitionMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AnnounceMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_EndOfGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AskKickDeniedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_EndOfHandHideCardsMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AskKickPlayerMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_EndOfHandShowCardsMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AuthClientResponseMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_ErrorMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AuthServerChallengeMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameAdminChangedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AuthServerVerificationMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListAdminChangedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AvatarDataMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListNewMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AvatarEndMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListPlayerJoinedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AvatarHeaderMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListPlayerLeftMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_AvatarRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListSpectatorJoinedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_ChatMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListSpectatorLeftMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_ChatRejectMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameListUpdateMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_ChatRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GamePlayerJoinedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_DealFlopCardsMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GamePlayerLeftMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_DealRiverCardMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameSpectatorJoinedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_DealTurnCardMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameSpectatorLeftMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82
Type_DialogMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82	Type_GameStartInitialMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 82

Type_GameStartRejoinMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_PlayersActionDoneMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_HandStartMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_PlayersTurnMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_InitAckMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_RejectGameInvitationMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_InitMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_RejectInvNotifyMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_InviteNotifyMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_RejoinExistingGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_InvitePlayerToGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_RemovedFromGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_JoinExistingGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ReportAvatarAckMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_JoinGameAckMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ReportAvatarMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_JoinGameFailedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ReportGameAckMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_JoinNewGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ReportGameMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_KickPetitionUpdateMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ResetTimeoutMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_KickPlayerRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_ShowMyCardsRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_LeaveGameRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_StartEventAckMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_MyActionRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_StartEventMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_PlayerIdChangedMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_StartKickPetitionMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_PlayerInfoReplyMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_StatisticsMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_PlayerInfoRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_SubscriptionRequestMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83
Type_PlayerListMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83	Type_TimeoutWarningMessage	(pokerth- proto.pokerth_pb2.PokerTHMessage attribute), 83

Type_UnknownAvatarMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	VOTEKICKREPLYMESSAGE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
Type_VoteKickReplyMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	voteKickReplyReceived()	(pokerthproto.protocol.PokerTHProtocol method), 112
Type_VoteKickRequestMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	VoteKickReplyType	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105
Type_YourActionRejectedMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	voteKickReplyType	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105
U			
unauthenticatedLogin	(pokerthproto.pokerth_pb2.InitMessage attribute), 58	VOTEKICKREPLYTYPE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105
unhandledMessageReceived()	(pokerthproto.protocol.PokerTHProtocol method), 112	VoteKickRequestMessage	(class in pokerthproto.pokerth_pb2), 105
UnknownAvatarMessage	(class in pokerthproto.pokerth_pb2), 103	voteKickRequestMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 88
unknownAvatarMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 87	VOTEKICKREQUESTMESSAGE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84
UNKNOWNAVATARMESSAGE_FIELD_NUMBER	(pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84	voteKickRequestReceived()	(pokerthproto.protocol.PokerTHProtocol method), 112
unknownAvatarReceived()	(pokerthproto.protocol.PokerTHProtocol method), 112	W	
unpack() (in module pokerthproto.transport), 113		WhichOneof()	(pokerthproto.pokerth_pb2.AdminBanPlayerAckMessage method), 12
unsubscribeGameList	(pokerthproto.pokerth_pb2.SubscriptionRequestMessage attribute), 102	WhichOneof()	(pokerthproto.pokerth_pb2.AdminBanPlayerMessage method), 13
V			
voteKick	(pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 106	WhichOneof()	(pokerthproto.pokerth_pb2.AdminRemoveGameAckMessage method), 14
VOTEKICK_FIELD_NUMBER	(pokerthproto.pokerth_pb2.VoteKickRequestMessage attribute), 106	WhichOneof()	(pokerthproto.pokerth_pb2.AdminRemoveGameMessage method), 15
voteKickAck	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105	WhichOneof()	(pokerthproto.pokerth_pb2.AfterHandShowCardsMessage method), 16
voteKickDeniedAlreadyVoted	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105	WhichOneof()	(pokerthproto.pokerth_pb2.AllInShowCardsMessage method), 18
voteKickDeniedInvalid	(pokerthproto.pokerth_pb2.VoteKickReplyMessage attribute), 105	WhichOneof()	(pokerthproto.pokerth_pb2.AllInShowCardsMessage.PlayerAllIn method), 17
VoteKickReplyMessage	(class in pokerthproto.pokerth_pb2), 104	WhichOneof()	(pokerthproto.pokerth_pb2.AnnounceMessage method), 19
voteKickReplyMessage	(pokerthproto.pokerth_pb2.PokerTHMessage attribute),		

WhichOneof() proto.pokerth_pb2.AnnounceMessage.Version method), 19	(pokerth-	WhichOneof() proto.pokerth_pb2.EndOfGameMessage method), 36	(pokerth-
WhichOneof() proto.pokerth_pb2.AskKickDeniedMessage method), 21	(pokerth-	WhichOneof() proto.pokerth_pb2.EndOfHandHideCardsMessage method), 37	(pokerth-
WhichOneof() proto.pokerth_pb2.AskKickPlayerMessage method), 22	(pokerth-	WhichOneof() proto.pokerth_pb2.EndOfHandShowCardsMessage method), 38	(pokerth-
WhichOneof() proto.pokerth_pb2.AuthClientResponseMessage method), 22	(pokerth-	WhichOneof() (pokerthproto.pokerth_pb2.ErrorMessage method), 39	(pokerth-
WhichOneof() proto.pokerth_pb2.AuthServerChallengeMessage method), 23	(pokerth-	WhichOneof() proto.pokerth_pb2.GameAdminChangedMessage method), 40	(pokerth-
WhichOneof() proto.pokerth_pb2.AuthServerVerificationMessage method), 24	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListAdminChangedMessage method), 41	(pokerth-
WhichOneof() proto.pokerth_pb2.AvatarDataMessage method), 25	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListNewMessage method), 42	(pokerth-
WhichOneof() proto.pokerth_pb2.AvatarEndMessage method), 26	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListPlayerJoinedMessage method), 43	(pokerth-
WhichOneof() proto.pokerth_pb2.AvatarHeaderMessage method), 27	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListPlayerLeftMessage method), 44	(pokerth-
WhichOneof() proto.pokerth_pb2.AvatarRequestMessage method), 27	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListSpectatorJoinedMessage method), 45	(pokerth-
WhichOneof() (pokerthproto.pokerth_pb2.ChatMessage method), 28	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListSpectatorLeftMessage method), 46	(pokerth-
WhichOneof() proto.pokerth_pb2.ChatRejectMessage method), 29	(pokerth-	WhichOneof() proto.pokerth_pb2.GameListUpdateMessage method), 46	(pokerth-
WhichOneof() proto.pokerth_pb2.ChatRequestMessage method), 30	(pokerth-	WhichOneof() proto.pokerth_pb2.GamePlayerJoinedMessage method), 47	(pokerth-
WhichOneof() proto.pokerth_pb2.DealFlopCardsMessage method), 31	(pokerth-	WhichOneof() proto.pokerth_pb2.GamePlayerLeftMessage method), 48	(pokerth-
WhichOneof() proto.pokerth_pb2.DealRiverCardMessage method), 32	(pokerth-	WhichOneof() proto.pokerth_pb2.GameSpectatorJoinedMessage method), 49	(pokerth-
WhichOneof() proto.pokerth_pb2.DealTurnCardMessage method), 33	(pokerth-	WhichOneof() proto.pokerth_pb2.GameSpectatorLeftMessage method), 50	(pokerth-
WhichOneof() proto.pokerth_pb2.DialogMessage method), 34	(pokerth-	WhichOneof() proto.pokerth_pb2.GameStartInitialMessage method), 51	(pokerth-
WhichOneof() proto.pokerth_pb2.EndKickPetitionMessage method), 35	(pokerth-	WhichOneof() proto.pokerth_pb2.GameStartRejoinMessage method), 53	(pokerth-

- WhichOneof() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinReplyData method), 73
- WhichOneof() (pokerthproto.pokerth_pb2.GameStartRejoinMessage.RejoinReplyData method), 53
- WhichOneof() (pokerthproto.pokerth_pb2.HandStartMessage method), 55
- WhichOneof() (pokerthproto.pokerth_pb2.HandStartMessage.PlainCards method), 55
- WhichOneof() (pokerthproto.pokerth_pb2.InitAckMessage method), 56
- WhichOneof() (pokerthproto.pokerth_pb2.InitMessage method), 57
- WhichOneof() (pokerthproto.pokerth_pb2.InviteNotifyMessage method), 58
- WhichOneof() (pokerthproto.pokerth_pb2.InvitePlayerToGameMessage method), 59
- WhichOneof() (pokerthproto.pokerth_pb2.JoinExistingGameMessage method), 60
- WhichOneof() (pokerthproto.pokerth_pb2.JoinGameAckMessage method), 61
- WhichOneof() (pokerthproto.pokerth_pb2.JoinGameFailedMessage method), 62
- WhichOneof() (pokerthproto.pokerth_pb2.JoinNewGameMessage method), 63
- WhichOneof() (pokerthproto.pokerth_pb2.KickPetitionUpdateMessage method), 64
- WhichOneof() (pokerthproto.pokerth_pb2.KickPlayerRequestMessage method), 65
- WhichOneof() (pokerthproto.pokerth_pb2.LeaveGameRequestMessage method), 66
- WhichOneof() (pokerthproto.pokerth_pb2.MyActionRequestMessage method), 67
- WhichOneof() (pokerthproto.pokerth_pb2.NetGameInfo method), 69
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerIdChangedMessage method), 70
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage method), 73
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData method), 73
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerInfoReplyMessage.PlayerInfoData.Avatar method), 72
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerInfoRequestMessage method), 74
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerListMessage method), 75
- WhichOneof() (pokerthproto.pokerth_pb2.PlayerResult method), 76
- WhichOneof() (pokerthproto.pokerth_pb2.PlayersActionDoneMessage method), 77
- WhichOneof() (pokerthproto.pokerth_pb2.PlayersTurnMessage method), 78
- WhichOneof() (pokerthproto.pokerth_pb2.PokerTHMessage method), 84
- WhichOneof() (pokerthproto.pokerth_pb2.RejectGameInvitationMessage method), 88
- WhichOneof() (pokerthproto.pokerth_pb2.RejectInvNotifyMessage method), 89
- WhichOneof() (pokerthproto.pokerth_pb2.RejoinExistingGameMessage method), 90
- WhichOneof() (pokerthproto.pokerth_pb2.RemovedFromGameMessage method), 91
- WhichOneof() (pokerthproto.pokerth_pb2.ReportAvatarAckMessage method), 92
- WhichOneof() (pokerthproto.pokerth_pb2.ReportAvatarMessage method), 93
- WhichOneof() (pokerthproto.pokerth_pb2.ReportGameAckMessage method), 94
- WhichOneof() (pokerthproto.pokerth_pb2.ReportGameMessage method), 95
- WhichOneof() (pokerthproto.pokerth_pb2.ResetTimeoutMessage method), 96
- WhichOneof() (pokerthproto.pokerth_pb2.ShowMyCardsRequestMessage method), 96
- WhichOneof() (pokerthproto.pokerth_pb2.StartEventAckMessage method), 97

WhichOneof() proto.pokerth_pb2.StartEventMessage method), 98	(pokerth-	yourActionRejectedReceived() proto.protocol.PokerTHProtocol 112	(pokerth- method),
WhichOneof() proto.pokerth_pb2.StartKickPetitionMessage method), 99	(pokerth-	yourAvatarHash proto.pokerth_pb2.InitAckMessage 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.StatisticsMessage 101	(pokerth- method),	YOURAVATARHASH_FIELD_NUMBER proto.pokerth_pb2.InitAckMessage 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.StatisticsMessage.StatisticsData method), 101	(pokerth-	yourPlayerId (pokerthproto.pokerth_pb2.InitAckMessage attribute), 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.SubscriptionRequestMessage method), 102	(pokerth-	YOURPLAYERID_FIELD_NUMBER proto.pokerth_pb2.InitAckMessage 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.TimeoutWarningMessage method), 103	(pokerth-	yourRelativeBet proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.UnknownAvatarMessage method), 104	(pokerth-	YOURRELATIVEBET_FIELD_NUMBER proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.VoteKickReplyMessage method), 105	(pokerth-	yourSessionId proto.pokerth_pb2.InitAckMessage 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2.VoteKickRequestMessage method), 106	(pokerth-	YOURSESSIONID_FIELD_NUMBER proto.pokerth_pb2.InitAckMessage 56	(pokerth- attribute),
WhichOneof() proto.pokerth_pb2>YourActionRejectedMessage method), 107	(pokerth-		
winnerPlayerId proto.pokerth_pb2.EndOfGameMessage attribute), 36	(pokerth-		
WINNERPLAYERID_FIELD_NUMBER proto.pokerth_pb2.EndOfGameMessage attribute), 36	(pokerth-		
wins (pokerthproto.game.Game attribute), 9			

Y

yourAction (pokerthproto.pokerth_pb2>YourActionRejectedMessage attribute), 107			
YOURACTION_FIELD_NUMBER proto.pokerth_pb2>YourActionRejectedMessage attribute), 107	(pokerth-		
yourActionRejected() proto.protocol.ClientProtocol method), 109	(pokerth-		
YourActionRejectedMessage (class in pokerth- proto.pokerth_pb2), 106			
yourActionRejectedMessage proto.pokerth_pb2.PokerTHMessage attribute), 88	(pokerth-		
YOURACTIONREJECTEDMESSAGE_FIELD_NUMBER (pokerthproto.pokerth_pb2.PokerTHMessage attribute), 84			