pin Documentation

Release 1

oleg

May 29, 2014

Contents

1 Social API 3

2 Profile API 7

3 Profile API 11

4 Indices and tables 13

Python Module Index 15

Contents:

CHAPTER 1

Social API

class `api.views.social.`**`AddCommentToBackground`**
API method that adds comment to background

class `api.views.social.`**`AddCommentToPhoto`**
API method that adds comment to photo

Link /api/social/photo/add_comment

`POST()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **comment** (*str*) – Comment body
- **photo_id** (*str*) – Id if the photo to comment
- **csid_from_client** (*str*) – CSID from client

Response data echoes given comment

class `api.views.social.`**`GetCommentsToBackground`**
API method that allows to get comments to background

class `api.views.social.`**`GetCommentsToPhoto`**
API method that allows to get comments to photo

class `api.views.social.`**`GetLikesToBackground`**
API method that allows to get likes to background

class `api.views.social.`**`GetLikesToPhoto`**
API method that allows to get likes to photo

class `api.views.social.`**`LikeDislikeBackground`**
API method that adds likes to background

class `api.views.social.`**`LikeDislikePhoto`**
API method that adds likes to photo

class `api.views.social.`**`LikeOrUnlikePin`**
Adds like to a certain pin.

Link /api/social/pin/like

`POST()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **csid_from_client** (*str*) – CSID from client
- **pin_id** (*str*) – Identifier of the pin

Response data returns status: success, if like was added

Example usage curl –data "csid_from_client=1&logintoken=RxPu7fLYgv&pin_id=46" http://localhost:8080/api/social/pin/like-unlike

class `api.views.social.`**`PostingOnUserPage`**

Provides sharing pins to social networks

Link /api/social/poup

`POST``()`

Share pins to social network Method PostingOnUserPage must receive next required params:

Parameters

- **share_list** (*str*) – list of pin's ids
- **access_token** (*str*) – access token for social network
- **social_network** (*str*) – name of social network

(for example 'facebook') :response data: list of pin ids to share, e.g. [10, 20, 30]

class `api.views.social.`**`QueryFollows`**

Class responsible for providing access to followers of a given user

Link /api/social/query/following

Link /api/social/query/followed-by

`POST``(`*query_type*`)`

Depending on query_type returns a list of users, following or followed by current user

Parameters

- **logintoken** (*str*) – Login token to authenticate current user
- **user_id** (*str*) – Quieried user id.

To test curl –data "csid_from_client=1&user_id=78&logintoken=RxPu7fLYgv" http://localhost:8080/api/social/query/following

To test curl –data "csid_from_client=1&user_id=78&logintoken=RxPu7fLYgv" http://localhost:8080/api/social/query/followed-by

Response data list of users

class `api.views.social.`**`SocialMessage`**

API method that sends message to user :link: /api/social/message

`POST``()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **content** (*str*) – Content of the message
- **user_id_list** (*str*) – List of user ids
- **csid_from_client** (*str*) – CSID from client

Response data empty

class `api.views.social.`**`SocialMessageToConversation`**
API method that sends message to conversation

Link /api/social/message_to_conversation

`POST``()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **content** (*str*) – Content of the message
- **user_id_list** (*str*) – List of conversation ids
- **csid_from_client** (*str*) – CSID from client

Response data empty

class `api.views.social.`**`SocialQueryConversations`**
API method that allows to get conversations

Link /api/social/query/conversations

`POST``()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **csid_from_client** (*str*) – CSID from client

Response data returns a list of conversations

class `api.views.social.`**`SocialQueryMessages`**
API method that allows to get messages from conversation

`POST``()`

Parameters

- **logintoken** (*str*) – Logintoken used fo authentication
- **conversation_id** (*str*) – Conversation id
- **csid_from_client** (*str*) – CSID from client

Response data list of all messages, for a given conversation

Profile API

class `api.views.profile.`**`ManageGetList`**

Link /api/profile/mgl

`POST``()`

Manage list of user products: sharing, add, remove

Method for image_id_share_list must additional receive next :param str csid_from_client: Csid string from client :param str logintoken: Logintoken :param str social_network: e.g. facebook :response_data: returns added/removed/shared getlists

class `api.views.profile.`**`SetPrivacy`**

Allows to set privacy level of the profile.

Link /api/profile/userinfo/update

`POST``()`

Updates profile with fields sent from the client, returns saved fields.

Parameters

- **csid_from_client** (*str*) – csid from client key
- **getlist_privacy_level/private** (*str*) – controls privacy level
- **logintoken** (*str*) – logintoken

Response_data Returns api response with 'getlist_privacy_level/private.'

To test curl –data "logintoken=UaNxct7bJZ&twitter=1&csid_from_client=1" http://localhost:8080/api/profile/userinfo/update

class `api.views.profile.`**`UserInfoUpdate`**

Defines a class responsible for updating user data, inside the profile

Link /api/profile/userinfo/update

`POST``()`

Updates profile with fields sent from the client, returns saved fields.

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **logintoken** (*str*) – Logintoken
- **<field>** (*str*) – The field which will be changed

Response_data returns changed field

To test curl –data "logintoken=UaNxct7bJZ&twitter=1&csid_from_client=1" http://localhost:8080/api/profile/userinfo/update

class `api.views.profile.`**`GetProfileInfo`**

Allows to render profile information, by token

Url /api/profile/userinfo/get

`POST``()`

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **logintoken** (*str*) – Logintoken

Response_data 'id', 'name', 'about', 'city', 'country','hometown', 'about', 'email', 'pic', 'website', 'facebook', 'twitter', 'getlist_privacy_level', 'private', 'bg', 'bgx', 'bgy', 'show_views', 'views', 'username', 'zip', 'linkedin', 'gplus', 'bg_resized_url', 'headerbgy', 'headerbgx'

To test curl –data "logintoken=UaNxct7bJZ&csid_from_client=123" http://localhost:8080/api/profile/userinfo/get

class `api.views.profile.`**`ProfileInfo`**

Returns public profile information

Url /api/profile/userinfo/info

`POST``()`

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **logintoken** (*str*) – Logintoken
- **username** (*str*) – Username
- **username** – id

Response_data 'id', 'name', 'about', 'city', 'country','hometown', 'about', 'email', 'pic', 'website', 'facebook', 'twitter', 'getlist_privacy_level', 'private', 'bg', 'bgx', 'bgy', 'show_views', 'views', 'username', 'zip', 'linkedin', 'gplus', 'bg_resized_url', 'headerbgy', 'headerbgx'

To test

- curl –data "csid_from_client=11&id=78&logintoken=RxPu7fLYgv" http://localhost:8080/api/profile/userinfo/info
- curl –data "csid_from_client=11&username=Oleg&logintoken=RxPu7fLYgv" http://localhost:8080/api/profile/userinfo/info

class `api.views.profile.`**`UpdateProfileViews`**

Responsible for updating count of pofile views

Link /api/profile/updateviews/<username>

`POST``(`*`profile`*`)`

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **profile** (*str*) – must be in url

Response_data returns a dict with 'status' key

To test curl –data "csid_from_client=11&logintoken=RxPu7fLYgv" http://localhost:8080/api/profile/updateviews/oleg

class `api.views.profile.`**`ChangePassword`**

Link /api/profile/pwd

`POST``()`

Change user password and get/store new logintoken :param str csid_from_client: Csid string from client :param str logintoken: Logintoken :param str old_password: current password of the user :param str new_password, new_password2: The new password typed 2 times

Response_data new clinet token

To test

class `api.views.profile.`**`QueryBoards`**

Class responsible for getting boards of a given user

Link /api/profile/userinfo/boards

`POST``()`

Returns all boards associated with a given user

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **user_id** (*str*) – id of the queried user

Response_data returns all boards of a given user as a list

To test curl –data "user_id=2&csid_from_client=1" http://localhost:8080/api/profile/userinfo/boards

class `api.views.profile.`**`QueryPins`**

Responsible for getting pins of a given user

Url /api/profile/userinfo/pins

`POST``()`

Returns all pins associated with a given user

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **user_id** (*str*) – id of the queried user

Response_data Returns list of pins of a current user

To test curl –data "user_id=78&csid_from_client=1" http://localhost:8080/api/profile/userinfo/pins

class `api.views.profile.`**`TestUsernameOrEmail`**

Checks if given username or email is already added to database. in case if a username

Link /api/profile/test-username

`POST``()`

Parameters

- **csid_from_client** (*str*) – Csid string from client
- **logintoken** (*str*) – username_or_email

Response data returns 'ok' or 'notfound'

To test curl –data "csid_from_client=1&username_or_email=oleg" http://localhost:8080/api/profile/test-username

CHAPTER 3

Profile API

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

Python Module Index

a

api.views.social, 3