pyprotobuf Documentation

Release 0.8

nate skulic

June 25, 2014

Contents

protocol buffers compiler.

Converts from .proto to:

- python protorpc
- Closure library externs
- Closure library goog.proto2

Project page: http://code.google.com/p/pyprotobuf

CHAPTER 1

Contents

1.1 Supported Languages

1.1.1 ProtoRPC

pyprotobuf can compile proto files to https://code.google.com/p/google-protorpc/.

Example

example.proto:

```
message Item {
  optional string aString = 1;
  optional int32 aNumber = 2;
  required string aRequiredString = 3;
  repeated string aRepeatedString = 4;
}
```

Run the command `protopy --format python example.proto` to generate:

```
from protorpc import messages

class Item(messages.Message):
    aString = messages.StringField(1)
    aNumber = messages.IntegerField(2)
    aRequiredString = messages.StringField(3, required=True)
    aRepeatedString = messages.StringField(4, repeated=True)
```

Using the protorpc's DateTimeField

To use protorpc's DateTimeField, you must `import "protorpc/message_types.proto"`. This proto file is included with pyprotobuf.

Then you can define fields with the `protorpc.DateTimeMessage` type.

For example:

```
import "protorpc/message_types.proto";

message Test {
```

```
    optional protorpc.DateTimeMessage datetime = 1;
}
```

Generates:

```
from protorpc import messages
import protorpc.message_types


class Test(messages.Message):
    datetime = protorpc.message_types.DateTimeField(1)
```

Defining custom field types

pyprotobuf supports the option `python_field_type` to define the `protorpc.messages.Message`'s `protorpc.messages.Field` type.

This can be used to handle (un)serialization of message types to language native types.

For example:

```
message DateTime {
    required int64 microseconds = 1;
    option python_field_type = "example.module.DateTimeField";
}


message Test {
    optional DateTime datetime = 1;
}
```

Generates:

```
from protorpc import messages
import example.module


class DateTime(messages.Message):
    microseconds = messages.IntegerField(1)


class Test(messages.Message):
    datetime = example.module.DateTimeField(1)
```

In this case, the `example.module.DateTimeField`` class (not defined) should be customized by converting the microseconds to return a python datetime and vice-versa.

1.1.2 Javascript Externs

Example

Input file (test.proto):

```
option javascript_package = "com.example";

message Item {
  optional string aString = 1;
```

```
  optional int32 aNumber = 2;
  required string aRequiredString = 3;
  repeated string aRepeatedString = 4;
}
```

Generated javascript externs(`protopy --format externs test.proto`):

```
/** @constructor */
com.example.Item = function(){};

/** @type {string} */
com.example.Item.prototype.aString;

/** @type {number} */
com.example.Item.prototype.aNumber;

/** @type {string} */
com.example.Item.prototype.aRequiredString;

/** @type {[string]} */
com.example.Item.prototype.aRepeatedString;
```

1.1.3 Javascript Closure library

1.1.4 Golang

1.2 Extending pyprotobuf

1.2.1 Developing a custom generator

Create a setup.py

Custom generators can be registered by creating a setup.py with the entry point *pyprotobuf.generators*. pyprotobuf will detect the generator and make it available as a format.

setup.py:

```
setup(
    # ...
    entry_points = '''
    [pyprotobuf.generators]
    custom = custom_generator
    '''
    # ...
 )
```

Create a generator module

Note: The CodeGenerator api is unstable.

custom_generator.py:

```
from pyprotobuf.codegenerator import CodeGenerator

class Generator(CodeGenerator):
    def generate_file(self, protonode, **kwargs):
        """ Custom ProtoNode generating logic

            :return: The compiled code
            :rtype: str
        """
        pass

__generator__ = Generator
```

The compiler AST

The compiler produces a tree in the form of:

- RootNode * PackageNode
  - FileNode * MessageNode * ...

FileNodes without packages declarations are placed in an unnamed PackageNode under the root.

1.3 API Documentation

1.3.1 Compiler

class pyprotobuf.compiler.**Compiler**

1.3.2 Nodes

class pyprotobuf.nodes.**ParseNode**

children = None

Type list[pyprotobuf.nodes.ParseNode]

parent = None

name = None

comment = None

A comment associated with this node.

Type pyprotobuf.nodes.CommentNode

get_file()

add_child(*c*)

get_children()

get_child(*index*)

get_full_typename()

add_dependency(*dep*)

Parameters **dep** (*pyprotobuf.nodes.ParseNode*) – The node to add a dependency to.

get_dependencies()

Return type list[pyprotobuf.nodes.ParseNode]

get_parents()

Return type list[pyprotobuf.nodes.ParseNode]

get_root()

get_full_name()

get_children_of_type(*node_class*)

Return type list[V <= T]

resolve_name(*name*)

has_option(*name*)

set_option(*name*, *value*)

get_option(**args*)

to_proto()

exception pyprotobuf.nodes.**InvalidChildTypeError**(*child*, *accepted_types*)

class pyprotobuf.nodes.**RootNode**

get_package(*name*)

add_child(*c*)

class pyprotobuf.nodes.**FileNode**

filename = None

package_name = None

get_imports()

class pyprotobuf.nodes.**CommentNode**

class pyprotobuf.nodes.**Package**

name = ‘’

add_child(*c*)

is_named()

class pyprotobuf.nodes.**PackageDefinition**

name = None

to_proto()

class pyprotobuf.nodes.**ServiceNode**

name = None

to_proto()

class pyprotobuf.nodes.**MethodNode**

name = None

request_type = None

response_type = None

class pyprotobuf.nodes.**MessageNode**

name = None

tostr(*depth*)

to_proto()

class pyprotobuf.nodes.**FieldDescriptorNode**

Properties: label: One of "repeated", "optional" or "required". number: The tag/number/id of the field. name: The name of the field. type: Can be a string (Enum of proto types), MessageNode or EnumNode.

label = None

number = None

name = None

type = None

class LabelType

REPEATED = 'repeated'

OPTIONAL = 'optional'

REQUIRED = 'required'

FieldDescriptorNode.**to_proto**()

class pyprotobuf.nodes.**EnumNode**

name = None

has(*key*)

get(*key*)

to_proto()

class pyprotobuf.nodes.**EnumAssignmentNode**

name = None

value = None

to_proto()

class pyprotobuf.nodes.**OptionNode**

name = None

value = None

to_proto()

class pyprotobuf.nodes.**ExtendNode**

name = None

message_node

Return type MessageNode

to_proto()

class pyprotobuf.nodes.**OptionalNode**

type = None

to_proto()

class pyprotobuf.nodes.**SyntaxNode**

to_proto()

class pyprotobuf.nodes.**ImportNode**

value: path

value = None

file_node = None

public = False

to_proto()

class pyprotobuf.nodes.**ExtensionsNode**

to_proto()

class pyprotobuf.nodes.**Types**

BOOL = 'bool'

STRING = 'string'

INT32 = 'int32'

INT64 = 'int64'

UINT32 = 'uint32'

UINT64 = 'uint64'

SINT32 = 'sint32'

SINT64 = 'sint64'

FIXED32 = 'fixed32'

FIXED64 = 'fixed64'

SFIXED32 = 'sfixed32'

SFIXED64 = 'sfixed64'

DOUBLE = 'double'

FLOAT = 'float'

BYTES = 'bytes'

ENUM = 'enum'

MESSAGE = 'messsage'

GROUP = 'group'

CHAPTER 2

Indices and tables

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

p