JSONRPC Documentation

Release 1.0

Edward Langley

March 29, 2016

Contents

Contents:

CHAPTER 1

Getting Started

This code will create a server that logs all requests, and provides two methods to clients: add and subtract:

```
#
#  Copyright (c) 2011 Edward Langley

from twisted.internet import reactor, ssl
from twisted.web import server
import traceback

from jsonrpc.server import ServerEvents, JSON_RPC

class ExampleServer(ServerEvents):
  # inherited hooks
  def log(self, responses, txrequest, error):
    print(txrequest.code, end=' ')
    if isinstance(responses, list):
      for response in responses:
        msg = self._get_msg(response)
        print(txrequest, msg)
    else:
      msg = self._get_msg(responses)
      print(txrequest, msg)

  def findmethod(self, method, args=None, kwargs=None):
    if method in self.methods:
      return getattr(self, method)
    else:
      return None

  # helper methods
  methods = set(['add', 'subtract'])
  def _get_msg(self, response):
    print('response', repr(response))
    return ' '.join(str(x) for x in [response.id, response.result or response.error])

  def subtract(self, a, b):
    return a-b

  def add(self, a, b):
    return a+b

root = JSON_RPC().customize(ExampleServer)
site = server.Site(root)
```

```

# 8007 is the port you want to run under. Choose something >1024
PORT = 8007
print('Listening on port %d...' % PORT)
reactor.listenTCP(PORT, site)
reactor.run()
```

To use this server (which is included as jsonrpc.example_server), start it and the client in this way:

```
% python -m jsonrpc.example_server &

% python -i -m jsonrpc.__main__ http://localhost:8007
```

```
>>> server.add(1,2)
3
>>> server.subtract(3,2)
1
>>> server.batch_call(dict(
...   add = ((3, 2), {} ),
...   subtract = ((), {'a': 3, 'b': 2})
... ))
[(5, None), (1, None)]
```

CHAPTER 2

JSON-RPC Server

CHAPTER 3

JSON-RPC Proxy

class jsonrpc.proxy.**JSONRPCProxy**(*host*, *path='jsonrpc'*, *serviceName=None*, **args*, ***kwargs*)
A class implementing a JSON-RPC Proxy.

Parameters

- **host** (*str*) – The HTTP server hosting the JSON-RPC server
- **path** (*str*) – The path where the JSON-RPC server can be found

There are two ways of instantiating this class: - JSONRPCProxy.from_url(url) – give the absolute url to the JSON-RPC server - JSONRPC(host, path) – break up the url into smaller parts

batch_call(*methods*)
call several methods at once, return a list of (result, error) pairs

Parameters **names** – a dictionary { method: (args, kwargs) }

Returns a list of pairs (result, error) where only one is not None

call(*method*, **args*, ***kwargs*)
call a JSON-RPC method

It's better to use instance.<methodname>(*args, **kwargs), but this version might be useful occasionally

classmethod from_url(*url*, *ctxid=None*, *serviceName=None*)
Create a JSONRPCProxy from a URL

class jsonrpc.proxy.**ProxyEvents**(*proxy*)
An event handler for JSONRPCProxy

Allow a subclass to do its own initialization, gets any arguments leftover from __init__

IDGen = '03f385e825fffbc462f2f433afe3105f655a4e6b'
an instance of a class which defines a __get__ method, used to generate a request id

get_params(*args*, *kwargs*)
allow a subclass to modify the method's arguments

e.g. if an authentication token is necessary, the subclass can automatically insert it into every call

proc_response(*data*)
allow a subclass to access the response data before it is returned to the user

CHAPTER 4

jsonrpc.jsonutil

jsonrpc.jsonutil.**encode**(*obj*, *skipkeys=False*, *ensure_ascii=True*, *check_circular=True*, *allow_nan=True*, *cls=None*, *indent=None*, *separators=None*, *encoding='utf-8'*, ***kw*)

Serialize obj to json, if it is not of a type which the encoder can handle, make it the proper type. Args and kw are as in json.dumps

jsonrpc.jsonutil.**decode**(*str*, *encoding=None*, *cls=None*, *object_hook=None*, *parse_float=None*, *parse_int=None*, *parse_constant=None*, ***kw*)

Return an object from a json string. This is just `json.loads()` renamed

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

j

B

C

D

E

F

G

I

J

P