

Bitcoin2Web2Py:

Bitcoin2Web2Py is a plugin which allows one to interface with the “Bitcoin” console application. It is fully functional, and easy to set up.

Note: the Bitcoin2Web2Py plugin has only been tested for Unix systems. So it may not work in windows.

Installing Bitcoin2Web2Py:

The first thing one must do when one wants to install Bitcoin2Web2Py is install Bitcoin. Bitcoin is a console application developed by the creators of Bitcoin. There is also a GUI version called bitcoin, and they are easily confused. Even if you have the GUI version of Bitcoin, you are going to have to install Bitcoin in order to communicate with Bitcoin2Web2Py.

The latest version of Bitcoin can be found here:

<http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.4.0/>

Home / Bitcoin / bitcoin-0.4.0

Name	Modified	Size	
Parent folder			
test	2011-09-16		
README.txt	2011-09-23	13.2 kB	
SHA1SUMS.asc	2011-09-23	582 Bytes	
bitcoin-0.4.0-src.tar.gz	2011-09-23	888.9 kB	
bitcoin-0.4.0-linux.tar.gz	2011-09-23	12.1 MB	
bitcoin-0.4.0-win32-setup.exe	2011-09-23	4.5 MB	
bitcoin-0.4.0-win32.zip	2011-09-23	7.8 MB	
bitcoin-0.4.0-macosx.dmg	2011-09-23	6.6 MB	
Totals: 8 Items		32.0 MB	

Select which version you need from this list, and install.

After downloading your desired version from the above website, install it.

Bitcoin2Web2Py uses a framework called Python JSON-RPC to communicate with Bitcoin. RPC stands for Remote Procedure Call, and this just what Python JSON-RPC does: It allows you to call Bitcoin procedures remotely though a secure internet connection. Go ahead and download JSON-RPC.

The latest version of Python JSON-RPC can be found here:

<http://json-rpc.org/wiki/python-json-rpc>

Installation

Download the source using bazaar.

```
$ bazaar checkout http://bazaar.json-rpc.org/trunk
```

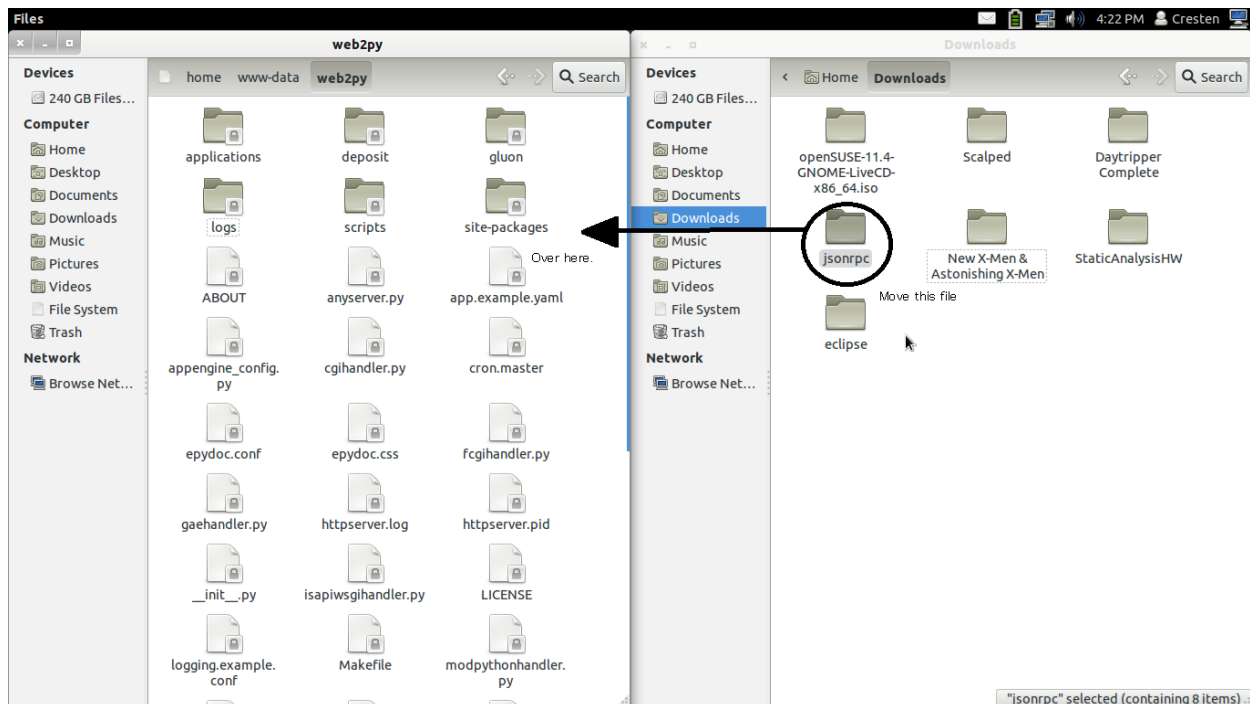
As root/administrator install the package running the provided setup script.

```
$ cd trunk/python-jsonrpc
$ python setup.py install
```

If you do not wish to install the package you can simply copy the `jsonrpc` folder to where python can find it when it searches for modules to be imported. E.g. this can be the same place where you python script resides in.

This image shows the section of the webpage that teaches you how to install JSON-RPC

Place the JSON-RPC file in the `/web2py/` directory, in the same folder as `web2py.py`.

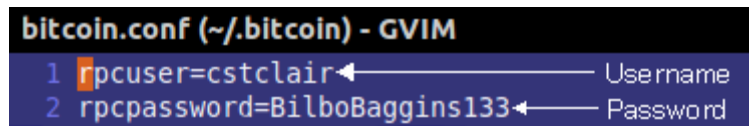


Now, go ahead and install the Bitcoin2Web2Py plugin. You are ready to configure your Bitcoin2Web2Py file!

Configure Bitcoin2Web2Py:

The first thing you're going to have to do is go to your (username)/.bitcoin folder, and edit your bitcoin.conf file. Create one if it does not exist.

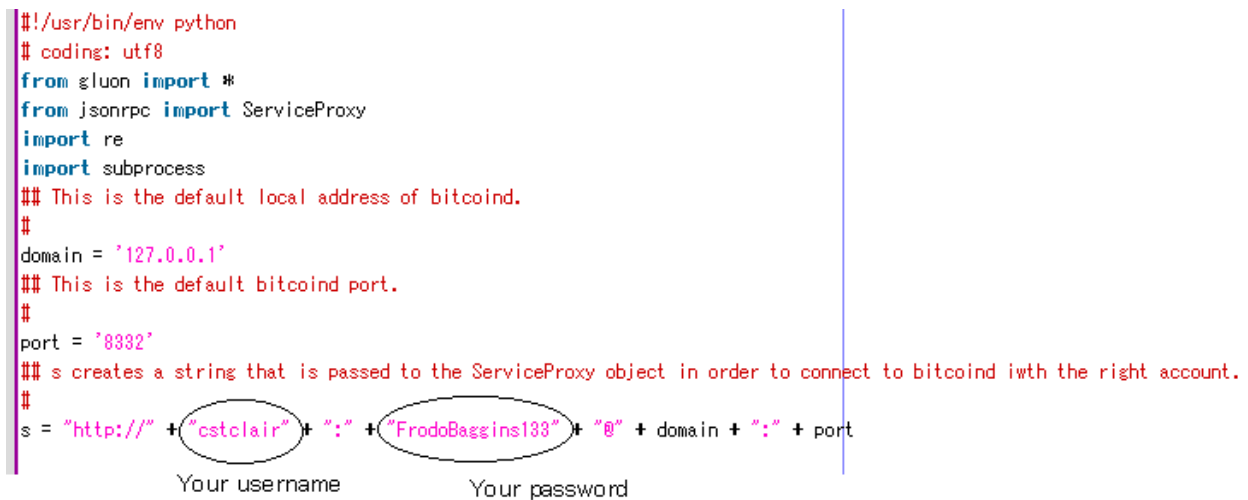
You will need to add 2 lines inside of this file:



```
bitcoin.conf (~/.bitcoin) - GVIM
1 rpcuser=cstclair ← Username
2 rpcpassword=BilboBaggins133 ← Password
```

Save the file and you're done with this folder!

Now, to log into this Bitcoin user, edit the plugin_bitcoin_client.py and put your chosen username and password into the indicated fields.



```
#!/usr/bin/env python
# coding: utf8
from gluon import *
from jsonrpc import ServiceProxy
import re
import subprocess
## This is the default local address of bitcoind.
#
domain = '127.0.0.1'
## This is the default bitcoind port.
#
port = '8332'
## s creates a string that is passed to the ServiceProxy object in order to connect to bitcoind with the right account.
#
s = "http://" + "cstclair" + ":" + "FrodoBaggins133" + "@" + domain + ":" + port

Your username      Your password
```

The above screenshot shows the variables you need to replace to access your Bitcoin username and password.

Starting Bitcoin2Web2Py:

By default, the code is set up to launch BitcoinD automatically when the controller for the webpage is called, and establishes a connection to the client through the JSON-RPC interface. This may be counterintuitive to how you plan on using Bitcoin2Web2Py, so in this section I will show you how to change this behavior.

Below is the code you will have to change to remove this behavior:

```
15
16 start_bitcoin() ← Spawn bitcoind instance
17 domain = '127.0.0.1'
18 print ('here')
19 port = '8332'
20 s = "http://" + "cstclair" + ":" + "BilboBaggins133" + "@"
21 access = ServiceProxy(s) ← Create JSON-RPC proxy object
22
```

Starting BitcoinD is always going to be the same. Call the `start_bitcoin()` method, and if an instance of BitcoinD is not already running, it will start. If you decide you want to change this behavior, you are going to have to take into account that Bitcoin2Web2Py's methods use the "access" object as a global object. If an instance of this object does not exist in global scope, the program will not work.

In order to create an access variable, you need a specifically formatted string as an argument. An image of the proper formatting is displayed below.

```
18 domain = '127.0.0.1' Default bitcoin address
19 port = '8332' Default bitcoin port
20 s = "http://" + "cstclair" + ":" + "BilboBaggins133" + "@" + domain + ":" + port
21 access = ServiceProxy(s) Bitcoin username
```