

RECONF(3)

LabRAD 使ってみる (1)

<https://github.com/labrad/pylabrad/wiki> を参考にするといいらしい

まずは Scalabrad <https://github.com/labrad/scalabrad> を用意

バイナリが <https://github.com/labrad/scalabrad/releases/tag/v0.8.3> で公開されている

```
wget -O scalabrad-0.8.3.tar.gz ¥
https://github.com/labrad/scalabrad/releases/download/v0.8.3/scalabrad-0.8.3.tar.gz
tar xvf scalabrad-0.8.3.tar.gz
./scalabrad-0.8.3/bin/labrad
```

これで scalabrad が起動する .

別のターミナルで Pylabrad を動かす .

まずは準備 .

```
pipenv --python 2.7
pipenv shell
pipenv install twisted
pipenv install pylabrad
```

python インタプリタを起動して ,

```
import labrad
cxn = labrad.connect('localhost') # ユーザ名とか聞かれるのでエンター
```

これでクライアントが起動した .

```
cxn
```

とすると

```
auth
manager
registry
```

というサーバーが起動していることが確認できる .

```
cxn['manager']; cxn.manager
```

で manager に接続 . たとえば , インスタンスの詳細を表示できる .

```
cxn.manager['data_to_string']; cxn.manager.data_to_string
```

でサーバの data_to_string が呼べるようになる . たとえば ,

```
cxn.manager.data_to_string([(1, 'This'), (2, 'is'), (3, 'a'), (4, 'test.')])
```

とかするとメソッドを呼び出せる .

LabRAD 使ってみる (2)

クライアントは <https://github.com/labrad/pylabrad/wiki/Writing-clients> に

サーバは <https://github.com/labrad/pylabrad/wiki/Writing-Servers> に詳しい説明がある .

が , まずは , サンプルとして

<https://raw.githubusercontent.com/wiki/labrad/pylabrad/squaringserver.py> を試してみる .

```
import time

from labrad.server import LabradServer, setting

class SquaringServer(LabradServer):
    name = "Squaring Server"

    @setting(10, data='v[]', returns='v[]')
    def square(self, c, data):
        time.sleep(2)
        return data**2

__server__ = SquaringServer()

if __name__ == '__main__':
    from labrad import util
    util.runServer(__server__)
```

このスクリプトを , pylabrad をセットアップ (pip install) した pipenv 環境で実行 .

```
python squaringserver.py
2020-01-24 20:07:32+0900 [-] Log opened.
```

みたいな表示が出力されるので , 一旦エンターキーを入力すると ,

```
Enter LabRAD password (localhost:7682):
```

とパスワードの入力を求められるので , 空のままエンター

```
2020-01-24 20:07:52+0900 [LabradProtocol,client] Squaring Server starting...
2020-01-24 20:07:52+0900 [LabradProtocol,client] Squaring Server now serving
```

とか出てサーバーが起動する .

別のターミナルでサーバにアクセスする .

```
pipenv shell
python
```

として , pylabrad をセットアップした環境に入って python インタプリタを起動 .

```
import labrad
cnx = labrad.connect('localhost')
# 'localhost' 不要という話もあるけどうちの環境では必要だった
```

これで ,

```
Enter username, or blank for the global user (localhost:7682):
Enter LabRAD password (localhost:7682):
```

と、ユーザ名とパスワード求められるので空のままエンター。
これで、squiringserver が待ち受けている LabRAD ネットワークに接続できる。

```
cnx
```

と入力すると、

```
Available servers:
  auth
  manager
  registry
  squaring_server
```

と、squiring_server の存在が確認できる。

```
s = cnx.squiring_server
```

と、squiring_server のインスタンスへのハンドラを変数に保存して、

```
s.square(2)
```

などを入力して squaring_server のメソッドが呼べる。
しばらく待つと、

```
4.0
```

と結果(二乗した値)がかえってくる。

LabRAD 使ってみる (3)

Python 3.8 でも、同様にして動作することを確認。

LabRAD 使ってみる (4)

リストをやりとりしたければ、

```
import time

from labrad.server import LabradServer, setting

class SquaringServer(LabradServer):
    name = "Squaring Server"

    @setting(10, "square")
    def square(self, c, data):
        time.sleep(2)
        return [data[0]**2, data[1]**2]

__server__ = SquaringServer()

if __name__ == '__main__':
    from labrad import util
    util.runServer(__server__)
```

とか用意しておいて、クライアントとなる側でインタプリタ起動して、

```
import labrad
cnx = labrad.connect('localhost'); cnx
s = cnx.squaring_server
s.square([2,3])
```

とすると、

```
array([4, 9], dtype=int32)
```

と返ってくる