

对于 virtualenv 报错 -----> 整个文件放在 C 盘 就可以

```
PS F:\1SMD\1代码项目\转pdf\calliope-lite-spreadsheet-master\server\services\fileserver> virtualenv ENV
usage: virtualenv [--version] [--with-traceback] [-v | -q] [--read-only-app-data] [--app-data APP_DATA] [--
y] [--creator {builtin,cpython2-win}] [--seeder {app-data,pip}]
      [--no-seed] [--activators comma_sep_list] [--clear] [--no-vcs-ignore] [--system-site-pack
...] [--pip version] [--wheel version] [--setuptools version]
      [--no-pip] [--no-wheel] [--no-setuptools] [--no-periodic-update] [--symlink-app-data] [--
dest
virtualenv: error: argument dest: invalid validate dest value: 'ENV'
SystemExit: 2
```

- virtualenv ENV

windows 下在 Scripts 文件夹而不是在 bin

- cd ../ENV/Scripts/
- activate

针对包安装不成功 一个包一个包安装

```
pip install -v flask==1.1.1 -i http://pypi.douban.com/simple/ --trusted-
host pypi.douban.com
```

- cd ../../
- cd server
- python run.py

运行后在生成数据事实时报错

```
PS F:\1SMD\1代码项目\转pdf\calliope-lite-spreadsheet-master\server\services\fileserver> virtualenv ENV
usage: virtualenv [--version] [--with-traceback] [-v | -q] [--read-only-app-data] [--app-data APP_DATA] [--
y] [--creator {builtin,cpython2-win}] [--seeder {app-data,pip}]
      [--no-seed] [--activators comma_sep_list] [--clear] [--no-vcs-ignore] [--system-site-pack
...] [--pip version] [--wheel version] [--setuptools version]
      [--no-pip] [--no-wheel] [--no-setuptools] [--no-periodic-update] [--symlink-app-data] [--
dest
virtualenv: error: argument dest: invalid validate dest value: 'ENV'
SystemExit: 2
```

需要修改代码

calliope-lite-spreadsheet-

master\server\services\generator\server\app\algorithm\RandomGenerator.py

注释的为原有代码 对其更改如下 (在提交时再修改回来!)

```
60 def generate(self):
61     # self.generateByParallel()
62     self.generateSequentially()
63
64     if len(self.generatedFacts) == 0 and self.random_policy == 'subspace':
65         self.random_policy = 'measure'
66         print('#####')
67         print('failed to generate in selected subspace: %s'%(self.selected_subspace))
68         print('#####')
69         print('switch random policy: %s'%(self.random_policy))
70         print('#####')
71         print('selected measure: %s'%(self.selected_measure[0]['field']))
72         print('#####')
73     # self.generateByParallel()
74     self.generateSequentially()
75
76     facts = list(self.generatedFacts)
```

```
123     candidates.sort(reverse=True, key=lambda x: x['score'])
124     self.processLock.acquire()
125     print('add %s random facts in %s'%(len(candidates),column))
126
127     # print('add %s random facts in %s'%(len(candidates),column['field']))
128     self.generatedFacts += candidates
```