Python flask web服务调用接口构建

# 一、文件组织目录



Backup\_dir 是备份文件目录，上传的文件会备份到此目录。

Dec\_dir 是解密目录，在这里生成解密文件，并返回。

Enc\_dir 生成加密后的文件，也就是.cpabe。

File\_dir 在这里生成公钥和私钥文件

Upload-dir 上传明文目录

Static：程序静态文件目录

Template：程序模板文件目录（flask）

App.py 主程序文件

CP\_ABE\_RUN.jar 程序依赖的Java 接口。

# 二、主程序app

## 1、程序依赖包:

Flask==1.1.1

Flask-Uploads==0.2.1

Flask-WTF==0.13.1

JPype1==0.7.0

Flask-Bootstrap==3.3.7.1

## 2、web配置信息

主要是一些文件系统路径的配置，以及文件上传的限制，以增强系统的安全性。

|  |  |
| --- | --- |
| pubfile | 公钥路径 |
| mskfile | 掩码路径 |
| prvfile | 私钥路径 |
| UPLOADED\_PHOTOS\_DEST | 上传文件的路径 |
| UPLOADED\_DECPRV\_DEST | 解密文件夹的路径 |
| UPLOADED\_BACKUP\_DEST | 备份文件夹的路径 |
| fileext | 限制上传文件扩展 |



在configure.uploads()中，使用[ , , ,]配置多个上传的路径。

## 3、一些form类

基于集成的flask\_WTF ,用来声明一系列的表单。

(FlaskForm):

|  |  |
| --- | --- |
| class name | function |
| UploadForm | 加密处上传文件表单 |
| UploadForm\_dec1 | 解密文件处上传私钥表单 |
| UploadForm\_dec2 | 解密文件处上传密文表单 |
| KeygenForm | 属性集合输入字符串输入表单 |
| InputForm | 加密授权策略输入表单 |

## 4、注册的路由

|  |  |
| --- | --- |
| route name | route page |
| / | home page 导航 |
| /setup | 初始化 函数调用界面 |
| /keygen | 产生私钥的界面接口 |
| /enc | 生成密文的加密接口 |
| /dec | 解密文件界面 |

## 5、路由函数

|  |  |  |
| --- | --- | --- |
| route name | function name | function |
| / | index\_page | 导航页 |
| /setup | setup\_page | 初始化页 |
| /keygen | keygen\_page | 私钥产生页 |
| /enc | enc\_page | 加密页 |
| /dec | dec\_page | 解密页 |

## 6、jpype 接口

|  |  |
| --- | --- |
| function name | function |
| init() | 启动jvm，返回Java Class |
| py\_setup() | Java api setup封装 |
| py\_keygen(attr\_str) | Java api 产生私钥封装 |
| py\_enc(policy, inputfile, encfile) | Java api 生成解密文件封装 |
| py\_dec(prvfile, encfile, decfile) | Java api 解密文件封装 |

## 7、主路由函数流程图

index\_page()



setup\_page()



keygen\_page()



enc\_page()



Dec\_page()



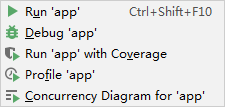
## 8、Java API calls

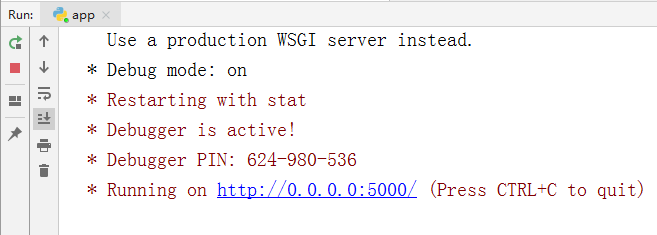
主要使用jpype依赖包，完成搭建在Java与python之间的桥梁，这是如何启动一个JVM虚拟机并完成Java API 调用的一个过程。



# 三、测试运行

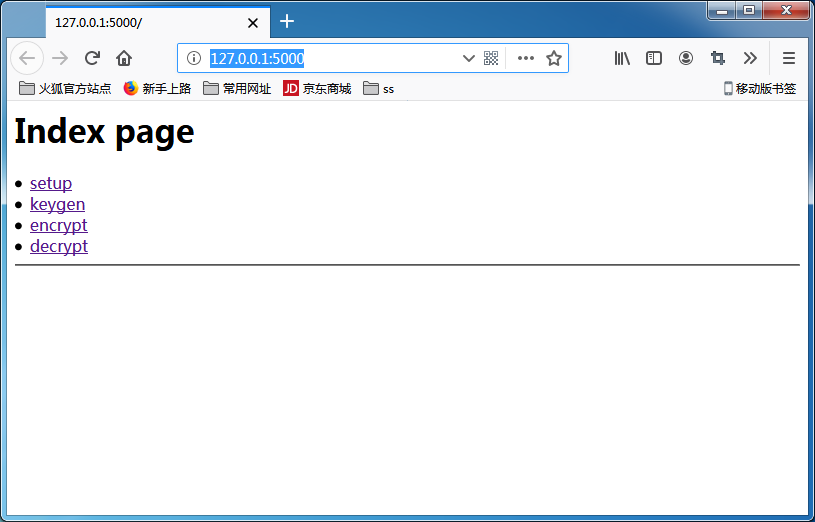
Run APP，python app.py

在pycharm 中，直接点击run app



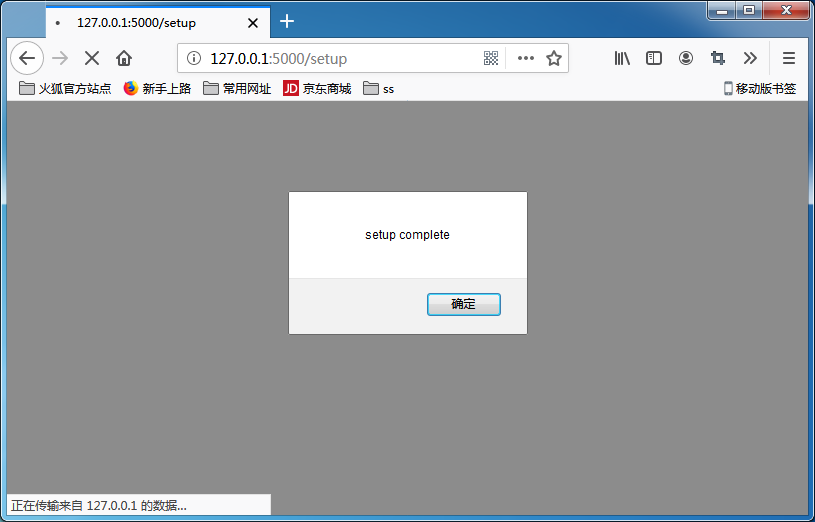
<http://127.0.0.1:5000/或者ip:5000>

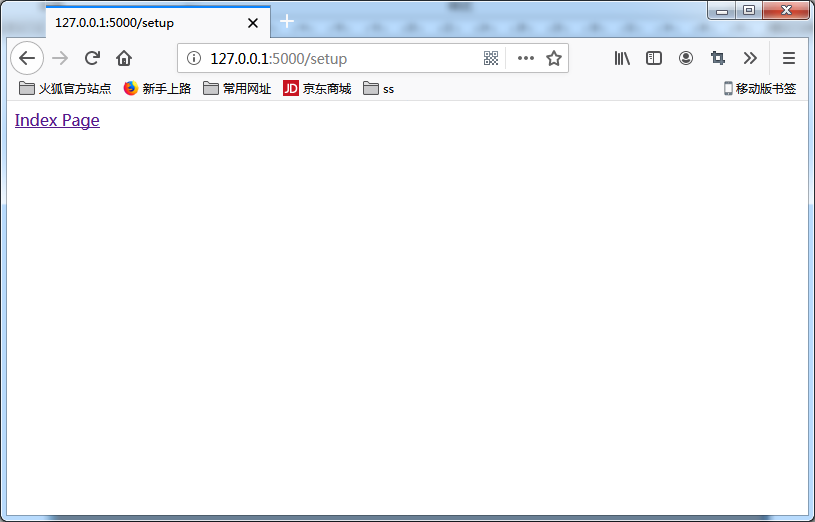
Home && Index Page



## 1、setup

点击setup,等待程序后台调用完毕，会显示run complete





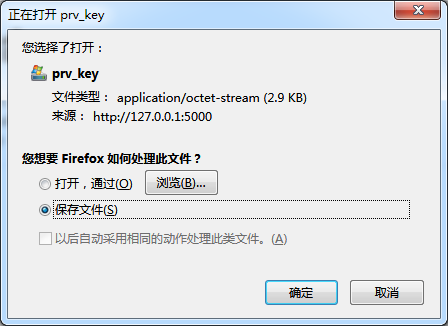
完成点击确定后根据导航回主页。

## 2、Keygen

输入属性集合：

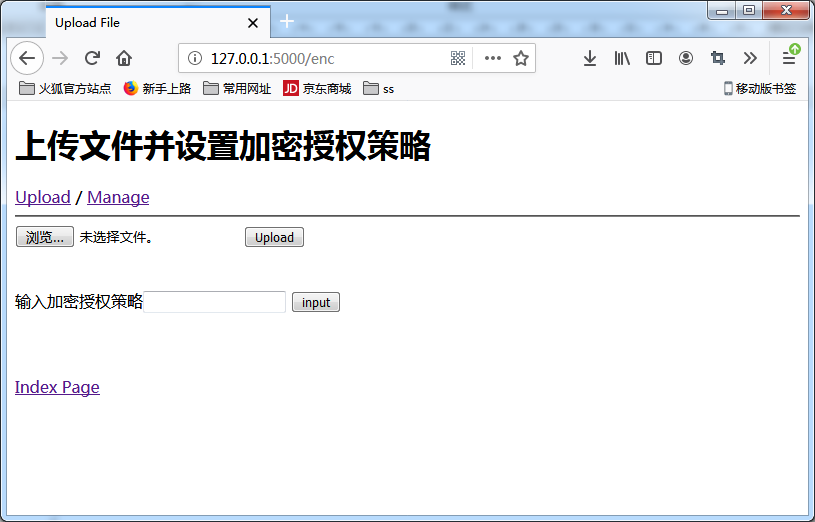


提交，会返回下载文件

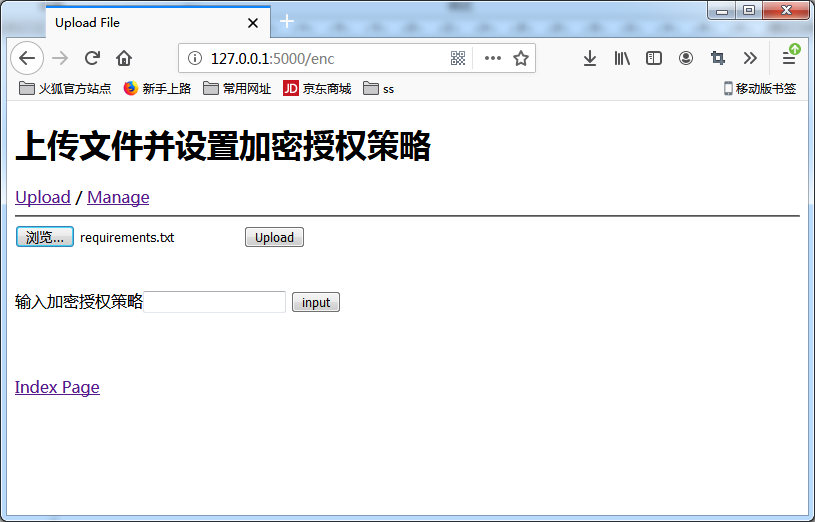


## 3、encrypted

点击encrypt

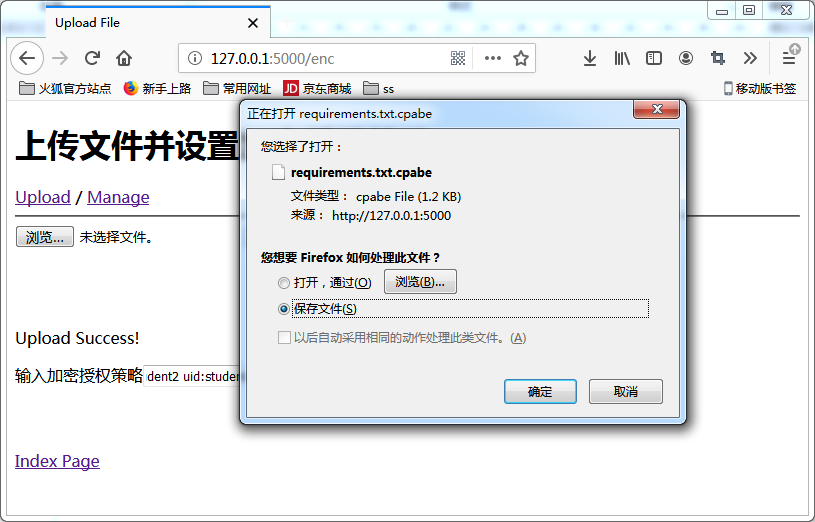


首先，上传加密文件



之后输入加密策略并提交



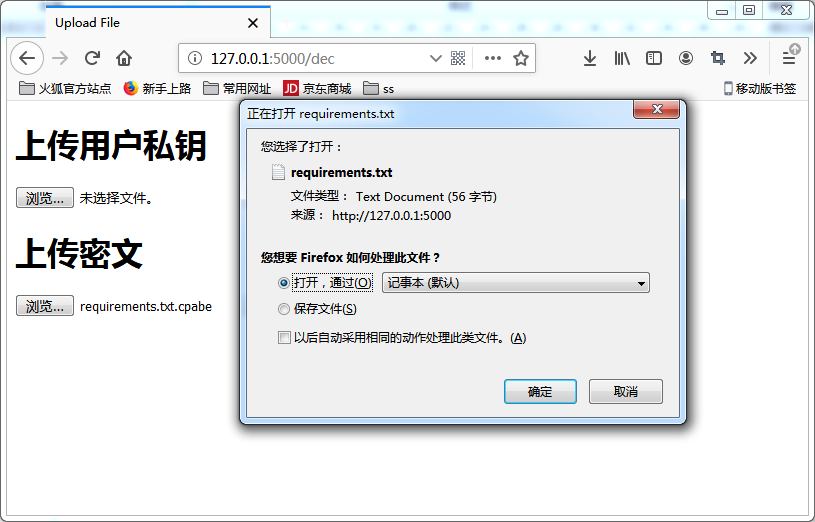


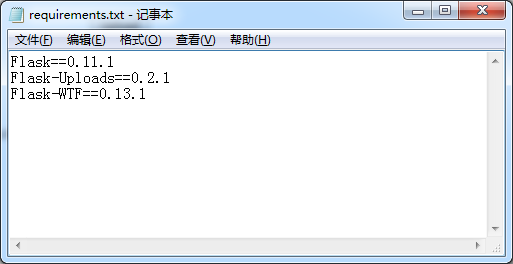
得到了加密后的密文。

## 4、decrypt

上传自己的私钥文件和待解密的密文文件。







完成解密，还原回来原来的文件。

这里要注意的是：

生成密文阶段，必须先上传明文文件，然后输入策略得到提交加密文件。

上传的密文必须是加密时生成的.cpabe扩展文件，否则不能上传，中间的原文件扩展名称须保留，否则系统无法生成对应的扩展，而不能得到正确的文件类型。

生成正确的文件需要扩展，否则虽然解密成功，但是须手动添加文件名称，正确的上传的文件名称是filename.extension.cpabe.