**项目流程与参数配置**

**//参数配置**

1，manage.py

Manage.py为应用主入口文件，其中包含从django引入的方法，值得注意的是，django结构语句中没有分号，用换行作为结束符。

2，第一个sealdesigner是整个项目，包含所有项目代码：

Seal：所有后台

ShoppingCart：购物车

Users：用户信息

Templates：所有静态网页

Webstatic：包含所有css，js，img和公用文件

Sealdesigner：与项目用的的一个包，必须有

1. 相当于司令部，所有规则，动作都在这执行
2. \_\_init\_\_.py是一个名字以下划线开头的空文件(不是所有都为空，可以写入数据)，但不能删，也不能改名
3. \_\_init\_\_.py为类似package的标识文件，它把一个包的文件作为一个接口提供一个引擎，一次性加载和引入某个项目的所有文件。
4. Setting.py是一个配置文件，用于配置django web应用。所谓的web服务器就是一个container(容器)

把创建的app加入到installed\_apps中，目的是告诉django逻辑代码的位置

Root\_urlonf定义了 URL以及用于处理这些 URL 的代码之间的映射关系,任何连接都必须通过url匹配

Database：数据库配置static\_url = ‘/static/’

1. urls.py 路由匹配机制，django最important的机制

<link href=’/webstatic/css/admin/stylesheets/all/css’>第一个斜杠表示在该项目下

Url(r^’Users/’,include(‘users.urls’))其中^表面匹配起始的地方，从^后根据正则表达式匹配，^后的两个逗号则标明起始和结束。Url中，逗号前面为标识，逗号后是一个方法

3,后台菜单

nav.html 为绝对静态网页，其中引用bootstrap样式

bootstrap是一个样式库，用于展示页面效果

4，补充

Cleaned-date :清除用户输入的无用空格，不需要判断时字段前面还是后面的空格

{{uf.as-table}}:将一个名为uf的模板变量的内容以一个table的形式展示出来

Ip地址中，127.0.0.1:8000中的127.0.0.1是本机回环测试地址

5，需要学习补充的：

1. 正则表达式
2. 代码结构优化
3. 模块抽象出来：能共享的共享

//登录项目流程

1，在浏览器中输入地址127.0.0.1:8000/Seal/Login(127.0.0.1为本地回环测试地址；8000为默认端口号，可改),为什么输入的Seal/Login而不是其他的呢，这是通过与项目同名的sealdesigner文件夹下的urls.py匹配后，匹配到url(r’^seal/’,include(‘seal.urls’))，得到seal下的所有路径，然后在seal中的urls.py再进行匹配，发现从seal下的views中引入login视图

2，login函数分析：

(1)如果请求的方式为post，

1)把post过来的值传给form表单并实例化赋值于uf变量，

2)调用form的is\_valid()方法，判断输入数据是否合法，

3)如果form对象不合法，则弹出“用户名或密码不能为空”否则用cleaned\_data取出用户输入的用户名和密码和验证码，进行下一步。

4)将验证码和cookies中获取的验证码进行大写对比，判断验证码是否正确，，如果不正确，弹出“验证码错误”信息

5)验证码正确的话，再从数据库中查询用户名并返回用户列表：

判断列表长度，如为空，弹出“用户名不存在”信息，否则进行下一步，

将用户输入密码和数据库储存用户密码进行对比，如错误，则弹出“密码错误”消息，否则进行下一步

6)将用户id存入浏览器cookies中，以做后续验证，然后跳转到/Seal/adminindex

如果为请求方式为get,则返回/Seal/login.html页面

2016.4.11