* Robolabel是一个的用于场景分类和物体标注的网站应用  
  后端用Python 3 写成，Web开发框架使用了Flask,使用了Flask模板引擎：Jinja2

数据存储用了sqlite和xml文件。

前端用了jquery, easy-autocomplete,bootstrap等

* 如果要启动一个本地开发服务器：

在easylabel文件夹下打开cmd,

输入

set FLASK\_APP=robolabel

flask run

然后在浏览器访问<http://127.0.0.1:5000/>

* 如果要把Flask 应用程序部署到服务器：

具体看这个教程

<http://www.pythondoc.com/flask/deploying/index.html>

* /mainpage页面用来场景分类

如果已经分类过一些图片，那么每次进入/mainpage页面都会从第一张未分类的图片开始

点击相应的分类按钮就会存储相应的分类在sql的名为post的TABLE中。

* 如何增加场景分类的类别

修改robolabel\templates\label文件夹里的mainpage.html

* /annotation页面用来标注物体

功能简述：画完Boudingbox，点击Set tag,再点击Validate & Get next Image。如果页面显示data has been sent那么服务器端会生成与图片名称一致的xml文件。

/annotation目前还有两个功能没有完成

1. difficult的前后端（目前生成的xml文件中全部默认difficult为0）
2. 每次打开页面，使页面自动跳转到第一张未标注的图片

/annotation有几个bug:

1. 每次新进入页面都需要点一下Validate & Get next Image 按钮才会能够画boundingbox
2. 标注完最后一张图片时如果点击Validate & Get next Image时仍旧会显示最后一张图片，如果此时再次标注并点击Validate & Get next Image，则会生成一个文件名为空的xml文件，里面是最后一张图片的标注

* 如何增加物体标注的类别

robolabel\static文件夹里有list\_of\_tags.json，打开写入即可以增加tag的类目

* 用户的登录与注册

由robolabel文件夹下的auth.py实现

目前用户的登录与注册均对访问网站的人开放。后期可以仅开放登录功能，由管理员注册。

用户登录后方可访问/mainpage和/annotation

* WEB安全

没有测试过网站防止Sql注入和XSS攻击的能力

包括用户登录与注册的页面也不能防止SQL注入。

后续需要增加防攻击功能

* 文件结构

图片放在robolabel\static\images文件夹下

其余的CSS和javascript文件放在robolabel\static文件夹下

* 配置环境

服务器端需要安装的包有：Flask,sqlite3.