

# Test2: Python & Tornado

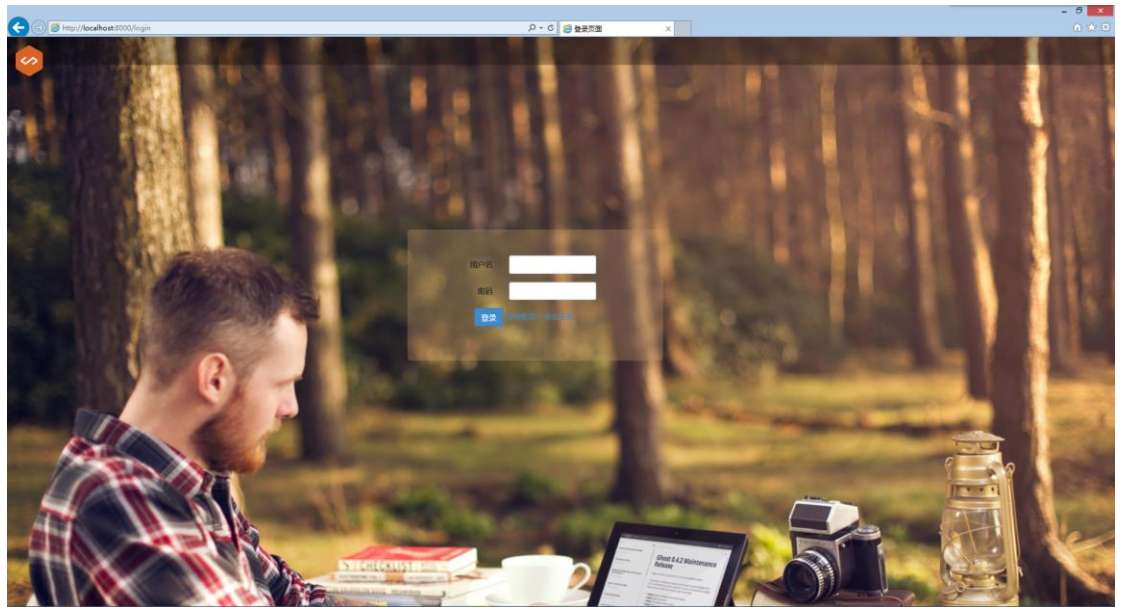
## 一、简述

同学们好，很高兴又能和大家一起享受美好的考试时光了，web2.0 这门课历年最好玩的就是它会有很多次的小考试。一来是检测大家对于这一部分知识点的掌握程度，二来是想要让大家熟悉这种编码的节奏，锻炼大家在短时间内完成编码任务的能力。

本次考试的重点是 Python 和 Tornado 的使用，下边的文档内容将会指导大家一步一步完成一个简单的小网站。在潘老师的要求下，我们已将静态页面提供给大家，所以在本次考试中大家不用编写前端 html 和 css，只用将我们提供的静态页面改成 tornado 模板即可使用。废话不多说，祝大家考试愉快。

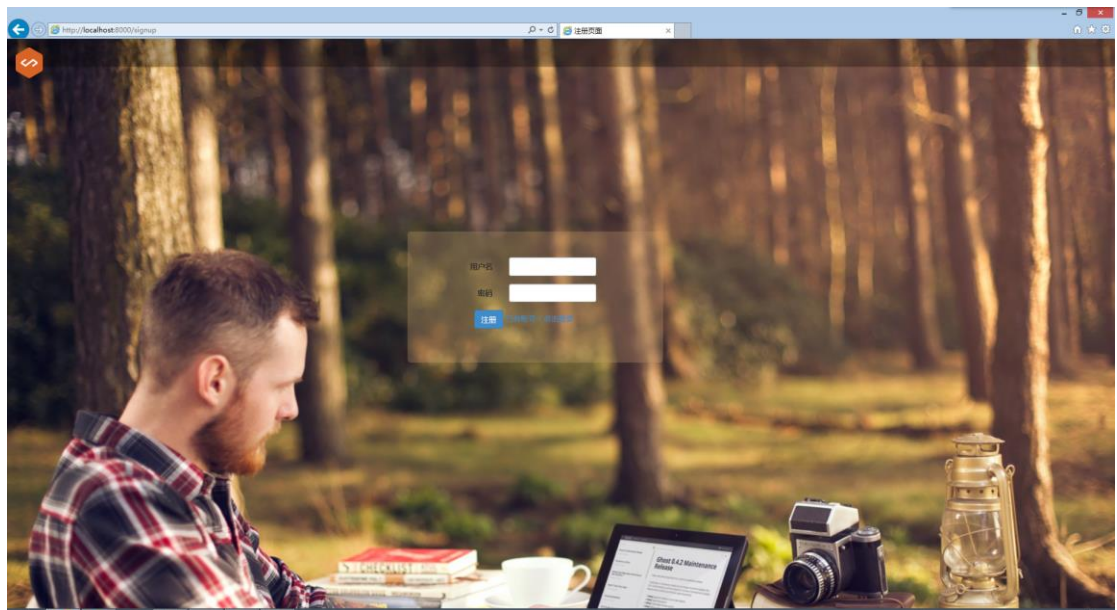
## 二、基本功能实现

### 1、 用户登录



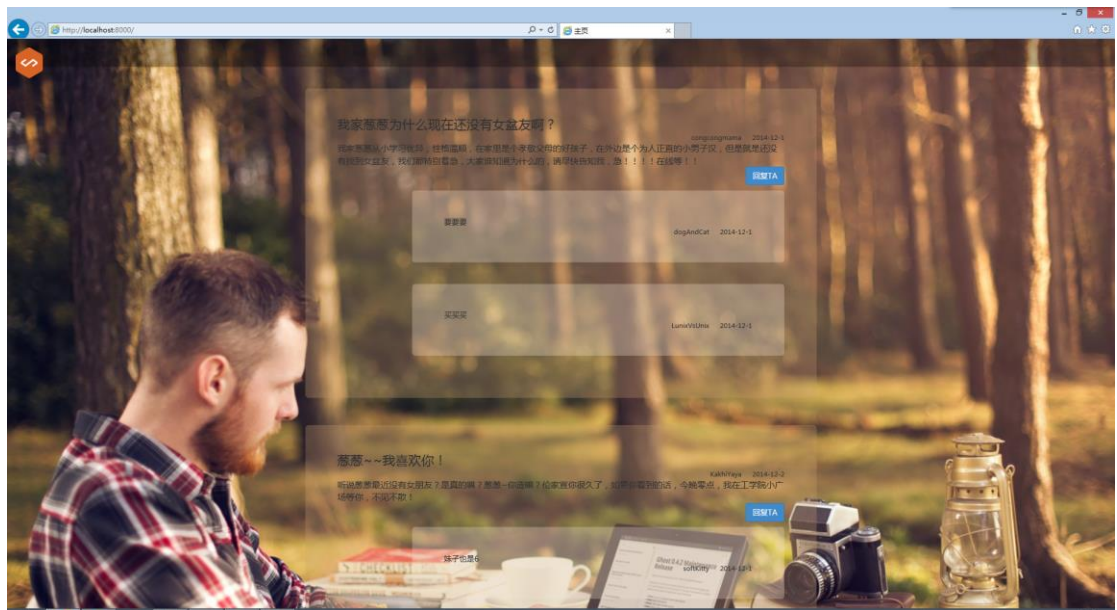
- ✦ url
  - ✧ `http://localhost:8000/login`
- ✦ 流程
  - ✧ 进入页面时，需要判断用户是否已经登陆（通过 cookie），若用户已经登陆，则直接跳转到主页；否则，继续留在当前页面
  - ✧ 用户填写用户名和密码，二者不能为空，否则登陆失败
  - ✧ 若都不为空，则读取 `userData.txt` 中的数据，匹配用户名和密码
  - ✧ 若匹配成功则表示登陆成功，否则表示登陆失败
  - ✧ 登陆成功后使用 cookie 保存用户信息并跳转到主页，登陆失败则继续留在当前页面
- ✦ 前后端数据交互
  - ✧ 页面标题“登陆页面”
  - ✧ 登陆按钮和注册页面链接，因为登陆和注册的静态页面为同一个，你要想办法通过交互数据显示正确的按钮和链接（并保证点击链接可以跳转到注册页面）

### 2、 用户注册



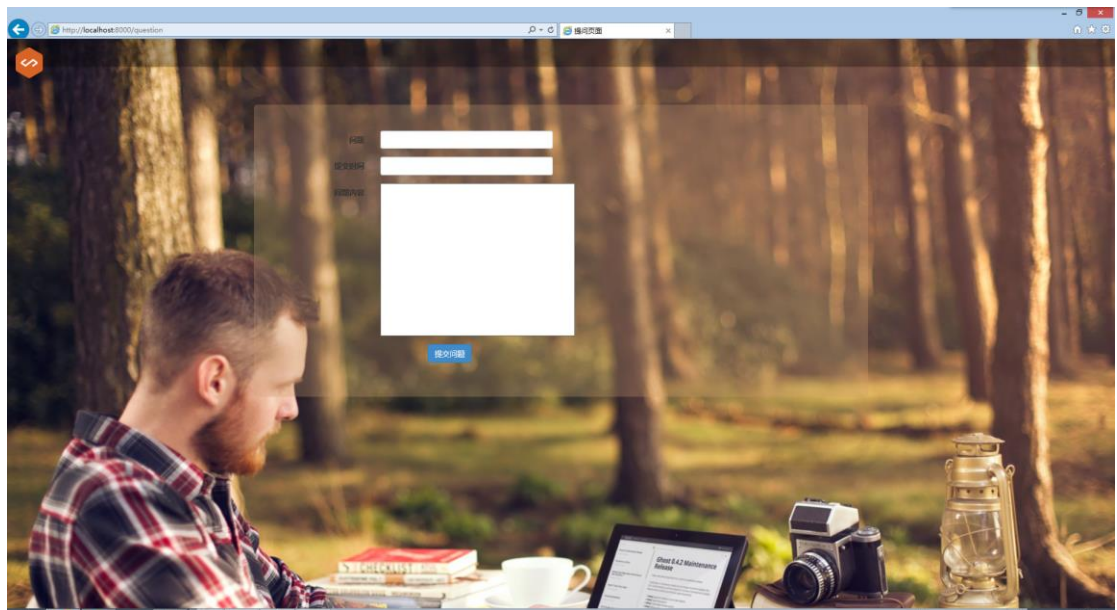
- ✦ url
  - ✦ `http://localhost:8000/signup`
- ✦ 流程
  - ✦ 进入页面时，需要判断用户是否已经登陆（通过 cookie），若用户已经登陆，则直接跳转到主页；否则，继续留在当前页面
  - ✦ 用户填写用户名和密码，二者不能为空，否则注册失败
  - ✦ 用户名的格式要求为 6-12 个英文字母（大小写均可）或数字，不允许出现其他字符；密码的格式要求为 6-12 个英文（大小写均可）或数字，不允许出现其他字符，密码开头必须是大写英文字母。如果格式不符合要求，注册失败。（这里会使用到 python 的正则表达式，可参考 <http://www.w3cschool.cc/python/python-reg-expressions.html>）
  - ✦ 用户名唯一，对于符合格式要求的用户名和密码，我们需要读取 `userData.txt` 中的数据，判断之前是否已经注册过相同的用户名，若用户名之前被注册过，注册失败
  - ✦ 前三步成功则表示注册成功
  - ✦ 注册成功后将用户名、密码写进 `userData.txt`（保存格式参考 `userData.txt` 中的示例），接着使用 cookie 保存用户信息并跳转到主页，注册失败则继续留在当前页面
- ✦ 前后端数据交互
  - ✦ 页面标题“注册页面”
  - ✦ 注册按钮和登陆页面链接，因为登陆和注册的静态页面为同一个，你要想办法通过交互数据显示正确的按钮和链接（并保证点击链接可以跳转到登陆页面）

### 3、显示主页面



- ✦ url
  - ✧ `http://localhost:8000/`
- ✦ 流程
  - ✧ 进入页面时，需要判断用户是否已经登陆（通过 cookie），若用户已经登陆，继续留在主页；否则，跳转到登录页面
  - ✧ 显示问题和回复，分别从 `questionData.txt` 和 `replyData.txt` 中读出数据，通过数据结构保存，渲染到前端
  - ✧ `questionData.txt` 中数据的格式为“标题；时间；作者；内容”；`replyData.txt` 中数据的格式为“回复对象；时间；作者；内容”
  - ✧ 我们要求，回复只能显示在回复对象的问题下边，根据回复对象（问题的作者）来安放回复
- ✦ 前后端数据交互
  - ✧ 页面标题“主页”
  - ✧ 问题和回复的数据。

#### 4、实现问题提交



- ✦ url
    - ✧ `http://localhost:8000/question`
  - ✦ 流程
    - ✧ 进入页面时，需要判断用户是否已经登陆（通过 cookie），若用户已经登陆，继续留在主页；否则，跳转到登录页面
    - ✧ 用户填写标题、时间（格式不限制）和内容，三者不能为空，否则提交失败
    - ✧ 三者都不为空时，将数据按照格式写到 `questionData.txt` 中
    - ✧ 提交成功，页面跳转到主页，否则继续留在当前页面
  - ✦ 前后端数据交互
    - ✧ 页面标题“提问页面”
    - ✧ 问题的数据：标题、时间和内容
- 5、用户登出
- ✦ url
    - ✧ `http://localhost:8000/logout`
  - ✦ 流程
    - ✧ 用户清除 cookie 数据，页面跳转到登录页面
  - ✦ 前后端数据交互
    - ✧ 无

### 三、附加功能实现

#### 1、实现对问题的回复

这个功能中，我们没有提供参考图，实现目标是点击“回复 TA”按钮，按钮下方出现输入框和提交按钮，可以输入信息（回复内容和时间），点击提交按钮，信息将被传回后端并写入 `replyData.txt` 中，页面刷新之后，可以看到刚刚的回复内容。实现方式不限，只要合情合理即可。（提示：可以使用 ajax。然后有的同学会问，这个 ajax 是个什么东东？这个，我说不清楚，有兴趣的同学自己去学习。反正是附加题嘛~就是挑战大神的题目啦~哇卡卡卡）

#### 2、404 页面

当用户访问非法的 url（例如 `http://localhost:8000/balabala`）时，我们将页面重定向到 404 页面，404 页面我们没有提供，自己写一个简单的就可以。



## 四、评分标准

### 1、基础分

- + 用户登录 10 分
- + 用户注册 30 分
- + 显示主页面 30 分
- + 实现问题提交 25 分
- + 用户登出 5 分

### 2、附加分

- + 实现对问题的回复 40 分
- + 404 页面 10 分
- + 良好的代码风格 5 分
- + 适量的注释 5 分
- + 模块化编程 5 分
- + 有 readme 文档 5 分

3、分数上限为 100 分，得分超过 100 的不再继续累加，发朵小红花好了

## 五、提交方式

### 1、提交地址

<ftp://my.ss.sysu.edu.cn/%7Epml/se805/2014fall/class-afternoon/week10-test/>

### 2、命名格式

学号\_姓名拼音\_版本号.zip，如 12330366\_yangliu\_v0.zip