课程介绍

课程介绍

- 项目搭建
- 数据库的设计与实现
- 项目模块化
- 页面功能开发

V1.0功能简介

- 完成首页关注的问题列表
- 完成问题详情

重点掌握

- 使用pyCharm创建及调试项目
- 数据库分析建模，ORM外键关联
- 使用蓝图来改进项目

项目搭建

项目搭建

- 使用pyCharm专业版来搭建实战项目
- 设计路由和视图函数
- 使用继承和包含语法实现模板结构的调整

数据库的设计与实现

功能分析

- 用户可以登陆及注册
- 用户可以发问题（标题、简述、内容、标签）
- 用户可以回答问题、关注问题
- 用户可以评论回答、点赞回答
- 个人中心

数据库的设计与实现

- pdman工具的使用

数据库的设计与实现

- 编写模型代码

项目模块化

思考

- 现在的实战项目有什么问题

蓝图（Blueprint）

什么是蓝图

- 大型应用组织管理
- 模块化/组件化
- 一个项目可以有多个蓝图

蓝图的实现方式

- 按功能划分
- 按模块划分

按功能划分

```
flasker/
    __init__.py
    static/
    templates/
        accounts/
        mall/
        mine/
    views/
        __init__.py
        accounts.py
        mall.py
        mine.py
    models.py
```

按模块划分

```
flasker/
    __init__.py
    accounts/
        __init__.py
        views.py
        static/
        templates/
    mall/
        __init__.py
        views.py
        static/
        templates/
    mine/
        __init__.py
        views.py
        static/
        templates/
    models.py
```

使用蓝图来改进项目

蓝图实现的过程

- 第一步：按模块拆分

ORM模型、配置、常量、工具类、功能模块等

蓝图实现的过程

- 第二步：视图文件中，实例化一个蓝图对象

```
accounts = Blueprint('accounts', __name__,
                     template_folder='templates',
                     static_folder='static')
```

蓝图实现的过程

◆ 第三步：注册蓝图

```
from accounts.views import accounts

app.register_blueprint(accounts, url_prefix='/accounts')
```

页面功能开发

页面功能开发

- 问题详情动态页面开发

  第一步：使用模板语法将详情页改造

  第二步：将需要展示评论、关注的信息从数据库取出

  第三步：取出第一条回答内容并展示剩余多少回答

页面功能开发

- 首页问题列表开发

页面功能开发

- 问题详情动态页面开发

课程总结

课程总结

- 知识点总结
- 踩坑指南

知识点总结

- 数据库模型的编写
- 外键引用，添加一对多属性
- 蓝图的使用

知识点总结

- 模板语法（标签、变量、过滤器）
- 模板继承，包含，本页面引用
- 自定义过滤器

踩坑指南

- 先建立数据库，再db.create_all()
- 使用蓝图后，db.create_all()的使用
- 扩展知识点：修改模型后的表同步(flask-migrate)