Django设计模式与最佳实践

目录

第一章：Django和设计模式

为什么是Django?

Django起源

新的框架的诞生

Removing the magic (扫除现有缺陷)

Django正在变得更好

Django是如何工作的？

什么是设计模式

Gang of four Patterns(四种设计模式)

Django符合MVC吗？

Fowler's Patterns(设计模式之花)

还有更多的设计模式吗？

本书中的设计模式

对设计模式的讨论

如何使用设计模式

最佳实践

Python之禅和Django设计哲学

本章总结

第二章：应用设计

如何收集需求

你会讲故事吗？（are you a story teller?）

HTML mockups

应用设计

将应用分解为App

复用还是自主开发（reuse or roll-your-own?）

我的应用沙盒

那个包来进行实现（which packages made it?）

在开启项目之前

超级之书--你的任务，你应该选择接受任务（SuperBook-your mission,should you choose to accept it）

为什么是Python3

开启项目

本章总结

第三章：模型（层）

M 比 V和C 更重要（M is bigger than V and C）

涉猎模型（层）（The model hunt）

将Models.py 分解成多个文件

结构模式（Structural patterns）

模式---标准化模型（normalized models）

问题描述

解决方案

模式---模型混入（model mixins http://www.zhihu.com/question/20778853）

问题描述

解决方案

模式--用户配置文件（user profiles）

问题描述

解决方案

模式----服务对象（service objects）

问题描述

解决方案

检索模式（retrieval patterns）

模式---属性字段（property field）

问题描述

解决方案

迁移（Migrations）

本章总结

第四章：视图和URLs

来自顶层的视图（a view from the top）

使视图变得优雅（Views got classier）

基于类的生成视图（class-based generic views）

视图混入（view mixins）

混入顺序

装饰器

视图模式

模式--读取可控视图（access controlled views）

问题描述

解决方案

模式--上下文强化（context enhancers）

问题描述

解决方案

模式----服务

问题描述

解决方案

设计URLs

URL剖析

在urls.py中发生了什么

URL模式语法

命名和命名空间

模型顺序（pattern order）

URL模型风格

本章总结

第五章：模版

理解Django模版语言特点

变量

属性

过滤器

标签

哲学--不要发明编程语言

组织模版

对其他模版语言的支持

使用Bootstrap

但是他们看起来一样

模版模式

模式--模版继承树（template inheritance tree）

问题描述

解决方案

模式--动态链接（the active link）

问题描述

解决方案

本章总结

第六章：管理接口（admin interface）

使用管理接口

为了便于管理强化模型（enhancing models for the admin）

并不是每个人都应该成为管理员

自定义管理接口

修改管理头部（changing the heading）

修改基栏和风格栏（changing the base and stylesheets）

为WYSIWYS 编辑添加富文本编辑器

Bootstrap-主题管理

彻底检修（complete overhauls）

管理权限保护

模式--特性标志位（feature flags）

问题描述

解决方案

本章总结

第七章：表单

表单是如何工作的

Django中的表单

为什么数据需要清理（why does data need cleaning）

展现表单（displaying forms）

time to crisp

理解csrf

采用基于类的视图处理表单（form processing with class-based views）

表单模式

模式---动态表单生成

问题描述

解决方案

模式--基于用户的表单

问题描述

解决方案

模式---每个视图下多表单动作（multiple form actions per view）

问题描述

解决方案

模式--CRUD 视图

问题描述

解决方案

本章总结

第八章：处理遗留代码

查询Django版本

激活虚拟环境

文件在哪里？这可不是PHP

从urls.py开始

跳出代码（jumping around the code）

理解代码基础（understanding the code base）

创建宏伟蓝图

增量式改变还是完全复写

在改变之前先进行测试

一步一步的编写测试代码

遗留数据库

本章总结

第九章：测试和Debug

为什么要测试

测试驱动的程序开发

写一个测试用例

采用断言的方法

编写更好的测试用例

嘲笑（mocking）

模式--通用测试和特例测试（test fixtures and factories）

问题描述

解决方案

了解测试的更多信息

调试（Debugging）

Django调试页面

更好的调试页面

print 函数

日志（logging）

Django 调试工具条

Python调试工具 pdb

其他调试

调试Django 模版

本章总结

第10章：安全性

跨页面脚本（xss）

为什么cookies是有价值的

django如何起作用

django不可用之处

cross-site request forgery(crsf)

django如何其作用

django不可用之处

sql注入

django如何起作用

django不可用之处

点击劫持（clickjacking）

django如何起作用

shell注入

django如何起作用

其他安全性列表

便捷的安全性列表

本章总结

第11章：生产部署（production-ready）

生产环境

选择网络协议（choosing a web stack）

协议栈组件（components of a stack）

集成（hosting）

作为服务平台（platform as a service）

虚拟私有服务器（virtual private servers）

其他的集成方法

部署工具

结构化部署（fabric）

典型的部署步骤

配置管理

监测

性能

前台性能

后台性能

模版

数据库

缓存

本章总结

附件 Python2 vs Python3

我还在使用Python2.7

Python3

用于Djangoauts 的Python3

将所有的\_\_unicode\_\_方法修改为\_\_str\_\_方法

所有类从积累Object类中继承

调用super()方法显得更加简单

相对导入必须明确（relative imports must be explictit）

HttpRequest 和HttpResponse 具有str和bytes 类型

Exception 语法进行了修正和提升

标准库重组

新的好处

采用Pyvenv 和Pip

其他改动

更多参考信息