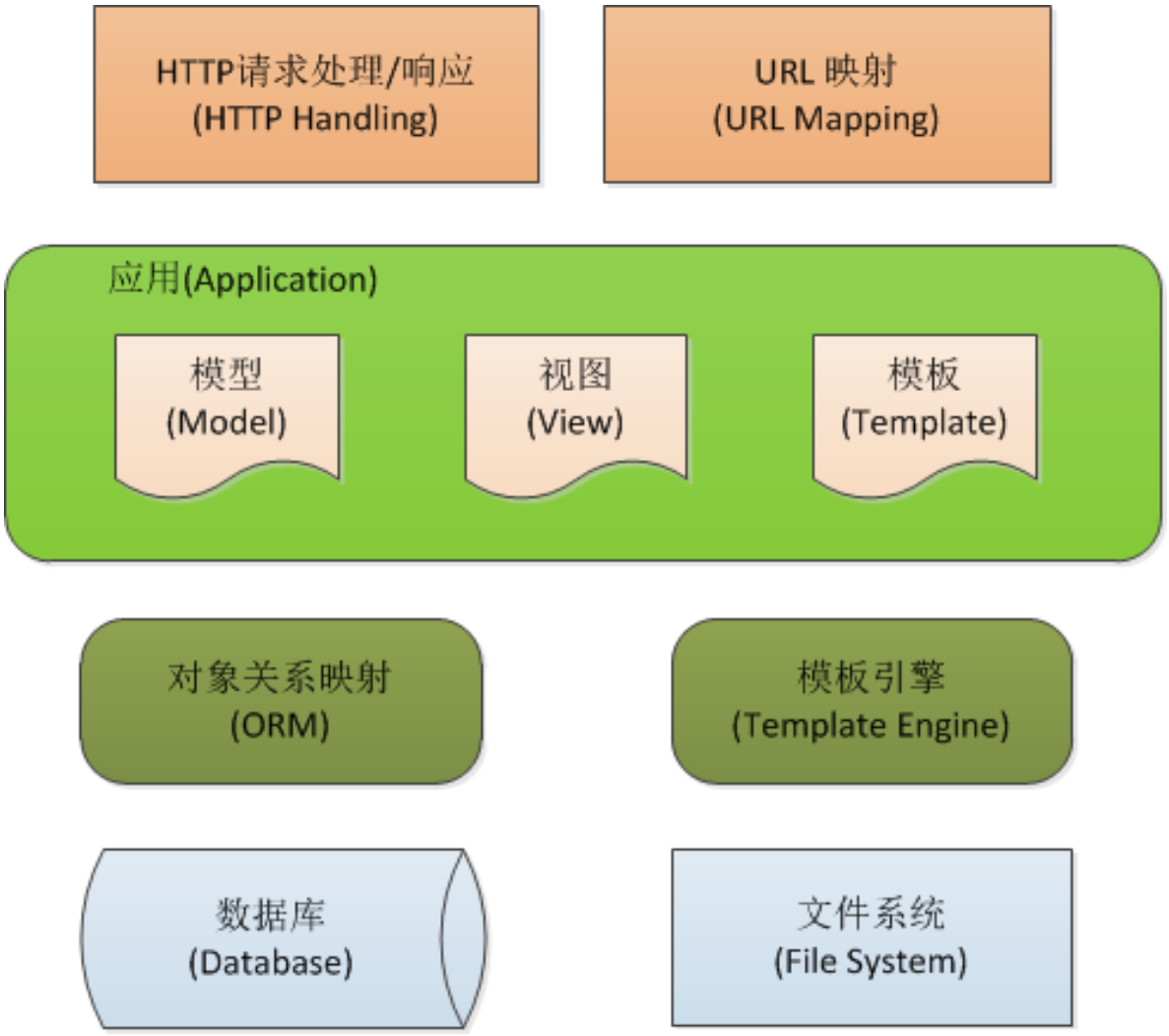
Django

- 袁永峰

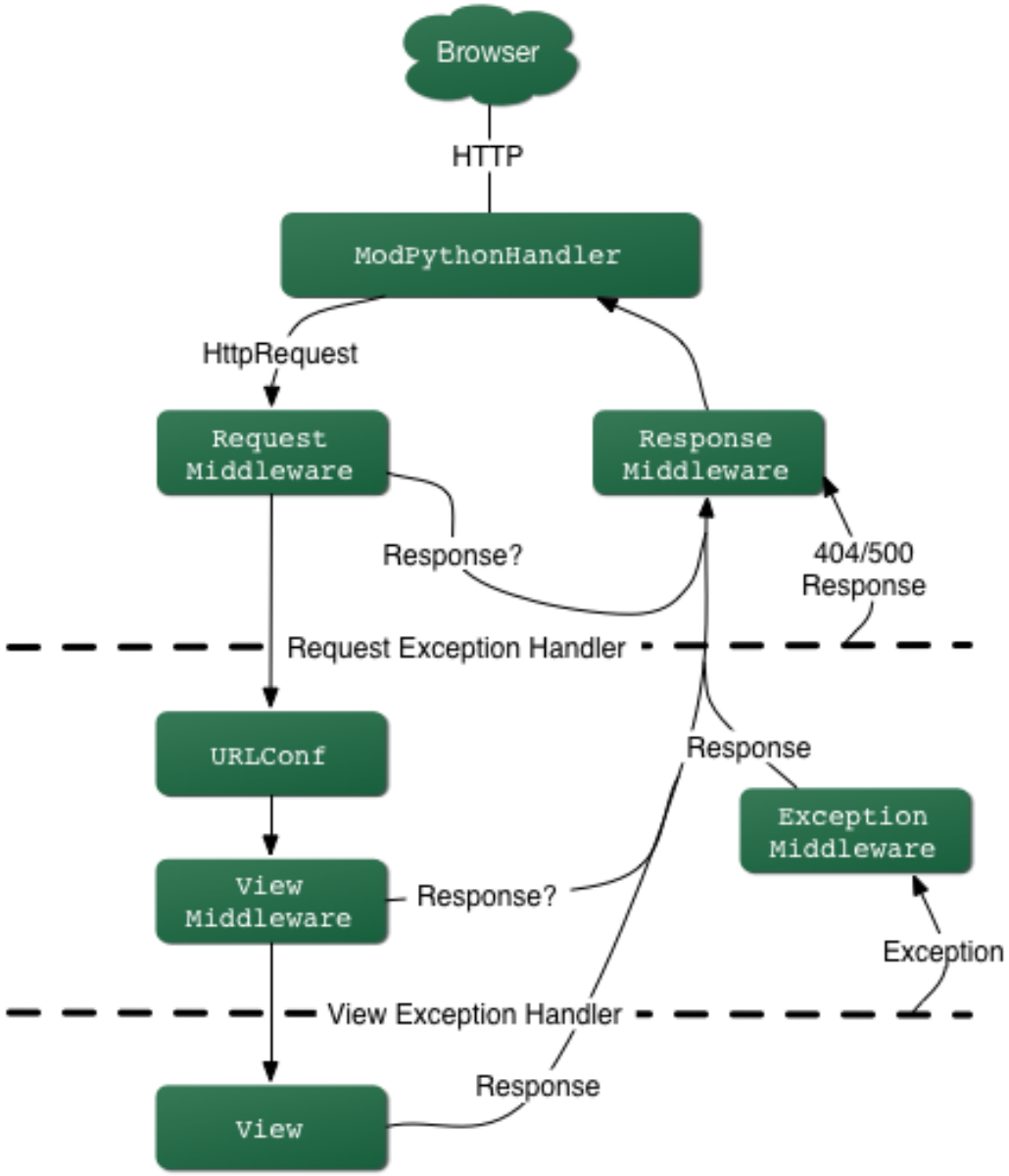
哈尔滨工业大学

Django架构总览



Django架构总览图

Django的中间件


Browser
HTTP
ModPythonHandler
HttpRequest
Request Middleware
Response Middleware
Response?
404/500 Response
Request Exception Handler
URLConf
Response
Exception Middleware
View Middleware
Response?
Exception
View Exception Handler
Response
View

1. 创建项目

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- 使用命令
  - python djangoadmin.py startproject myproject
- 获得帮助
  - python djangoadmin.py startproject –help

2. 运行开发服务器

- 使用命令
  - python manage.py runserver
- 获得帮助
- python manage.py runserver help
- 访问地址
  - http://127.0.0.1：8000
  - 可以看到一个默认的欢迎页面

3. 创建应用

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- 使用命令
  - python manage.py startapp myapp
- 获得帮助
  - python manage.py startapp –help

4. 设置数据库

- 编辑 settings.py, 指定数据库引擎、数据库名等配置项目
- 配置开发数据库
  - DATABASE_ENGINE = 'sqlite3'
  - DATABASE_NAME = 'myproject.db'
- 也可以设置为
  - 'postgresql_psycopg2', 'postgresql', 'mysql','ado_mssql'

5. 激活应用

- 编辑 settings.py ，添加应用的 package, 将使服务器启动时自动加载应用

```
INSTALLED_APPS = (
        'django.contrib.auth',
        'django.contrib.contenttypes',
        'django.contrib.sessions',
        'django.contrib.sites',
        'myproject.myapp',
)
```

6. 配置 URL 映射

- 每当 Django 收到 Request, 它将根据 URL 映射的模式 (URL pattern) 匹配到视图 (views.py) 中的回调函数 .

- Django 从上到下逐个进行模式匹配，当遇到第一个匹配的模式后停止，调用相应的视图方法处理 Request.

- 编辑项目目录下的 urls.py

增加 myapp 的 URL 映射

Django 使用正则表达式匹配 URL, 配置应用的映射内容如下：

```
from django.conf.urls.defaults import *
    urlpatterns = patterns('',
        (r'^entry/$', 'myapp.views.entry_list'),
        (r'^entry/(?P<object_id>\d+)/$', 'myapp.views.entry_detail'),
)
```

7. 使用 Admin Site

- **编辑项目根目录下的 urls.py ，去掉 admin 前的注释符号**
  - from django.conf.urls.defaults import *
  - # Uncomment the next two lines to enable the admin:
  - from django.contrib import admin
  - admin.autodiscover()
  - urlpatterns = patterns('',
  - # My App:
  - (r'^myapp/', include('myproject.myapp.urls')),
  - # Uncomment the next line for to enable the admin:
  - (r'^admin/(.*)', admin.site.root),)

7. 使用 Admin Site

- 使用命令 python manage.py syncdb

- 将自动创建 Admin Site 所需要的数据表，并可以
- 创建管理员帐号
- 再次运行开发服务器
- 访问
- http://servername:port/admin/
- 可以创建 User 和 Group 并设置权限

8.定义 Model

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- Model 是关系型数据在应用程序中的映射．通常数据库中的表和 Model 是相互对应的．表中的栏位表现为Model 中的属性

- 对于不需要存储在数据库表中的信息，可以通过定义一个叫作 Meta 的 Model 的内类 (Inner Class) 来实现

- Django 提供了通过 Model 自动生成数据库表和字段 ,并且访问操作数据的功能

8.定义 Model

- 编辑 myapp 目录下的 models.py 定义 Model

```
from django.db import models
class Entry(models.Model):
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateTimeField()
```

8.定义 Model

- 使用命令
- python manage.py shell
- >>>from django.db import models
- >>>dir(models)

定义 Fields & Relations

- 列举 fields:

- AutoField
- BooleanField
- harField
- CommaSeparatedIntegerField
- DateField
- DateTimeField
- DecimalField

EmailField

FileField <New> 使用 max_length 改变默认的 varchar(100)

FilePathField <New> 使用 max_length 改变默认的 varchar(100)

FloatField <Change> 表现为 Python 中的 float

IPAddressField

ImageField <New> 使用 max_length 改变默认的 varchar(100)

IntegerField

NullBooleanField

PhoneNumberField

- PositiveIntegerField
- PositiveSmallIntegerField
- SlugField
- SmallIntegerField
- TextField
- TimeField
- URLField
- USStateField
- XMLField
-

9. 定义 Fields & Relations

- **Django 支持的第二种 Model 继承类型是继承体系中的每一个 Model 都在数据库中建立自己的表，父类和子类通过一个自动建立 OneToOneField 关联**

```
class Place(models.Model):
    name = models.CharField(max_length=50)
    address = models.CharField(max_length=80)

class Restaurant(Place):
    serves_hot_dogs = models.BooleanField()
    serves_pizza = models.BooleanField()
```

9. 定义 Fields & Relations

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

```
class Person(models.Model):
        name = models.CharField(max_length=128)
        def __unicode__(self):
            return self.name
class Group(models.Model):
        name = models.CharField(max_length=128)
        members = models.ManyToManyField(Person, through='Me
    mbership')
        def __unicode__(self):
            return self.name
 class Membership(models.Model):
        person = models.ForeignKey(Person)
        group = models.ForeignKey(Group)
        date_joined = models.DateField()
        invite_reason = models.CharField(max_length=64)
```

10. 定义 Form

- Form 表单是字段 (Field) 的集合，能够验证数据，生成模板所需要的 HTML
- Field 字段是一个负责做数据验证的类
- Widget 协助字段生成 HTML
- Form Media 定义表单所需要的 CSS 和 Javascript 等资源

- Form 定义举例

```
from django import forms
 class ContactForm(forms.Form):
        subject = forms.CharField(max_length=100)
        message = forms.CharField()
        sender = forms.EmailField()
        cc_myself = forms.BooleanField(required=False)
```

- Form 输入 HTML, 使用 as_p(),as_table(),as_li()

```
>>>form.as_p
   <form action="/contact/" method="POST">
   <p><label for="id_subject">Subject:</label>
<input id="id_subject" type="text" name="subject" maxlength="100" /></p>
   <p><label for="id_message">Message:</label><input type="text" name="message" id=
"id_message" /></p>
   <p><label for="id_sender">Sender:</label><input type="text" name="sender" id="id_s
ender" /></p>
   <p><label for="id_cc_myself">Cc myself:</label><input type="checkbox" name="cc_my
self"
id="id_cc_myself" /></p>
   <input type="submit" value="Submit"></form>
```

11. 开发 Views

- Django 提倡 Model Template View 的开发方式
- Views 是编写的处理业务逻辑和表单提交的函数

- myproject

```
| myapp
    | __init__.py #  表明目录是一个 Python package
    | models.py   #  模型定义文件，根据模型定义生成数据库结构
    | views.py    #   视图方法定义
    | forms.py    #   表单定义
    | admin.py    #   定义 ModelAdmin
```

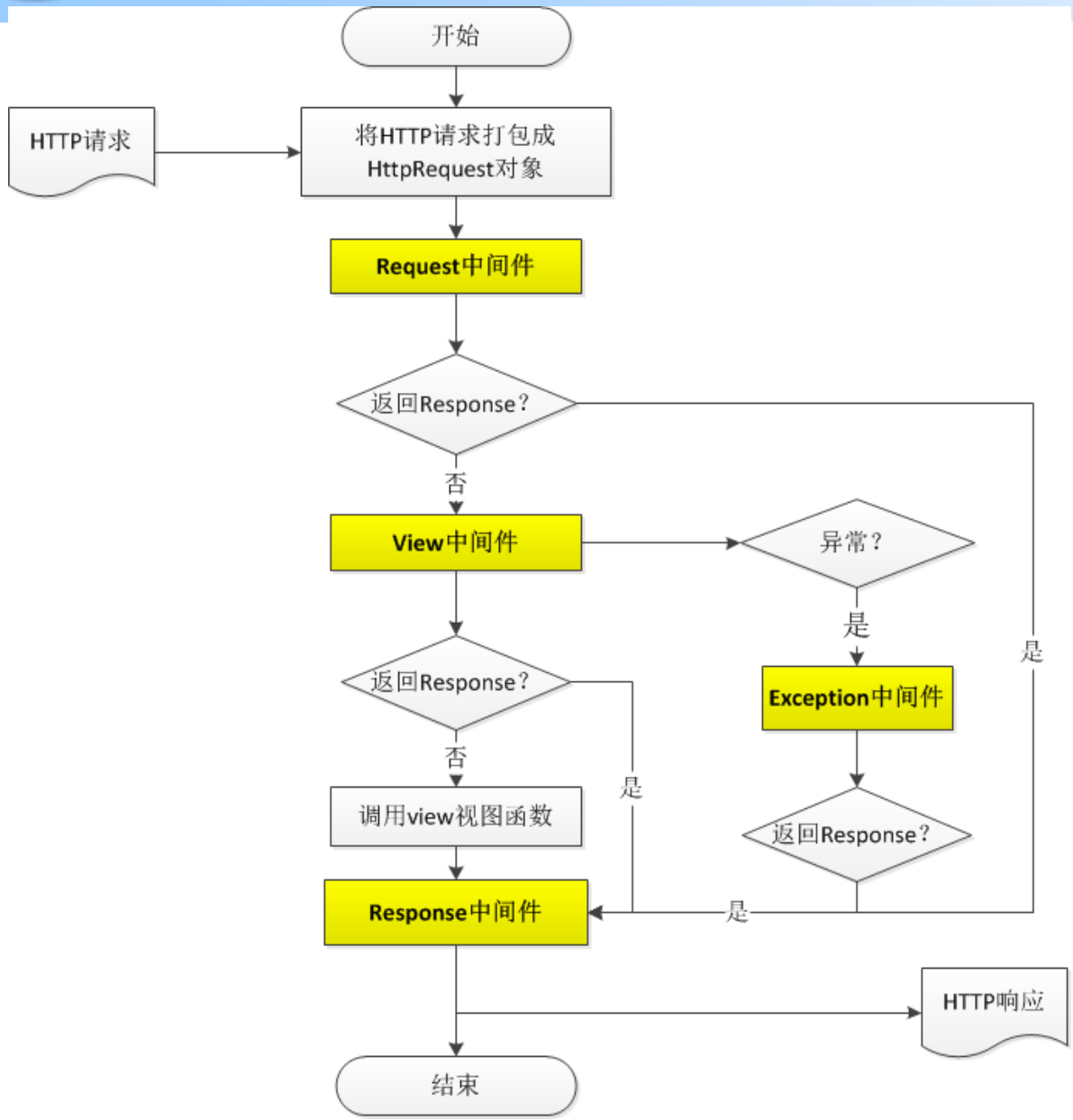
12.编写 Template

哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- 编辑项目目录下的 settings.py

```
TEMPLATE_DIRS = (
"/home/my_username/mytemplates", # Chan
ge this to your own directory.)
```

模板可以继承 {% extends "base.html" %}


开始
HTTP请求
将HTTP请求打包成
HttpRequest对象
Request中间件
返回Response?
否
View中间件
异常?
是
是
返回Response?
Exception中间件
否
是
调用view视图函数
返回Response?
Response中间件
是
HTTP响应
结束