蓝鲸智云 BLUEKING IT大咖说 知识共享平台

蓝鲸智云 腾讯云

Django基础

Django基础-MVC

蓝鲸智云讲师 jeremy

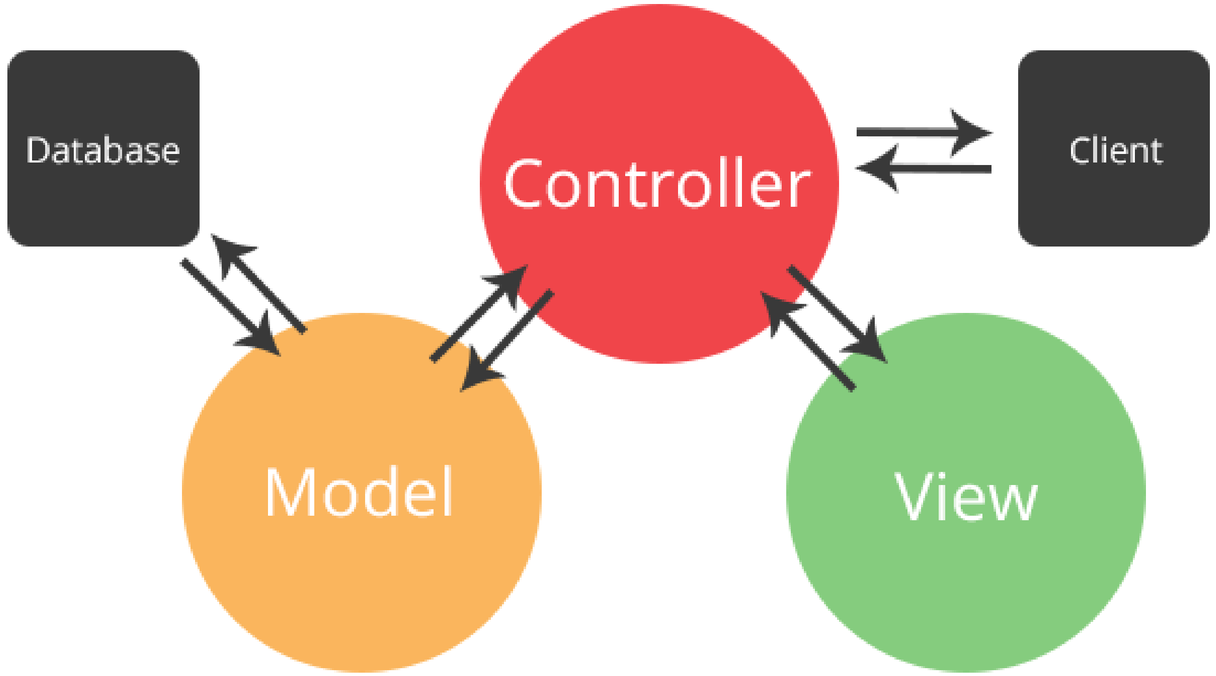
Django基础（一）

- MVC介绍
- Django框架结构
- Django-settings
- Django-URLConf
- Django-Views
- Request、Response对象介绍
- 登录验证和用户信息获取
- Pycharm调试

MVC介绍

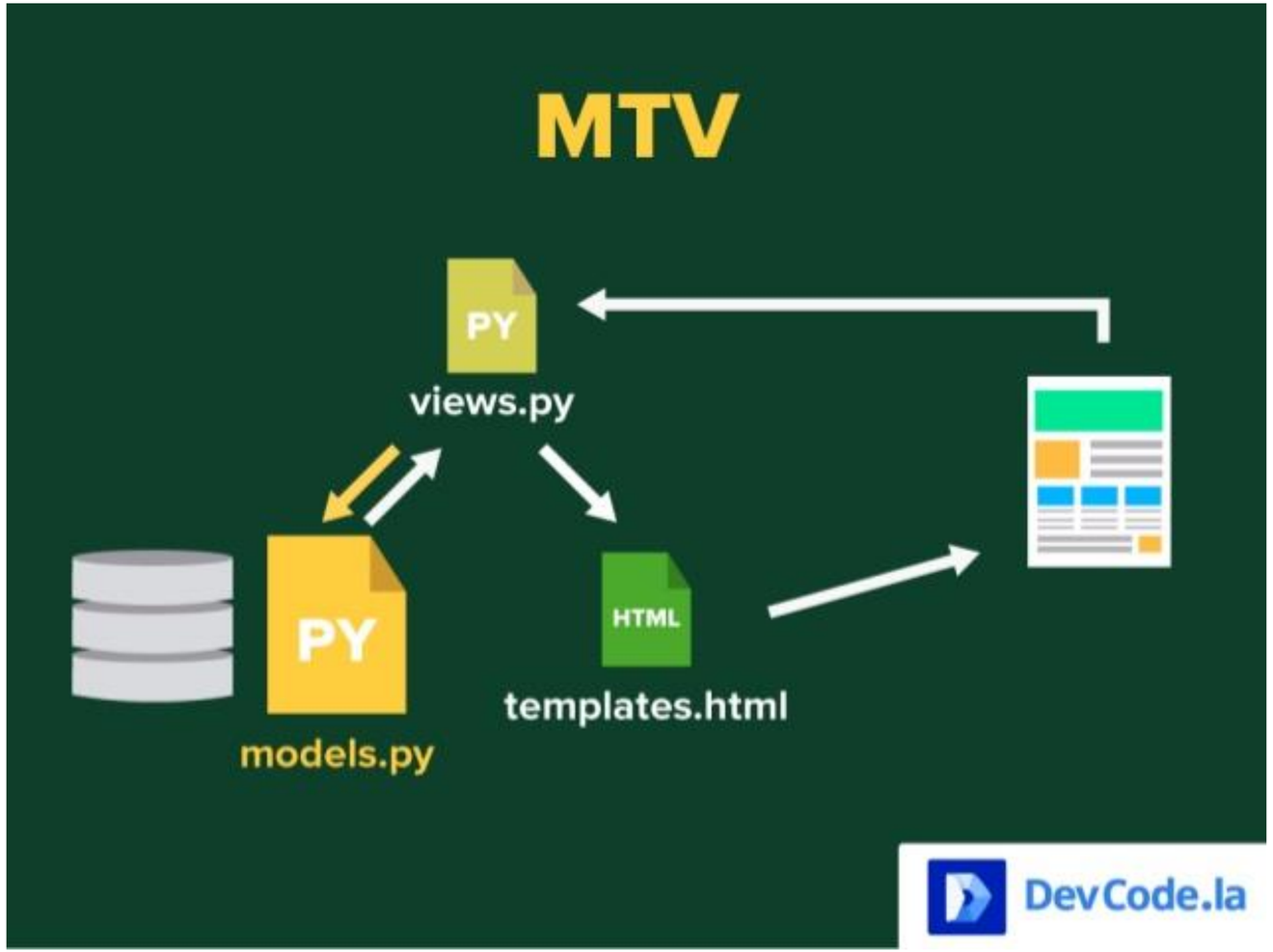
MVC

- MVC模式（Model–View–Controller）



Django框架结构 & settings

Django框架

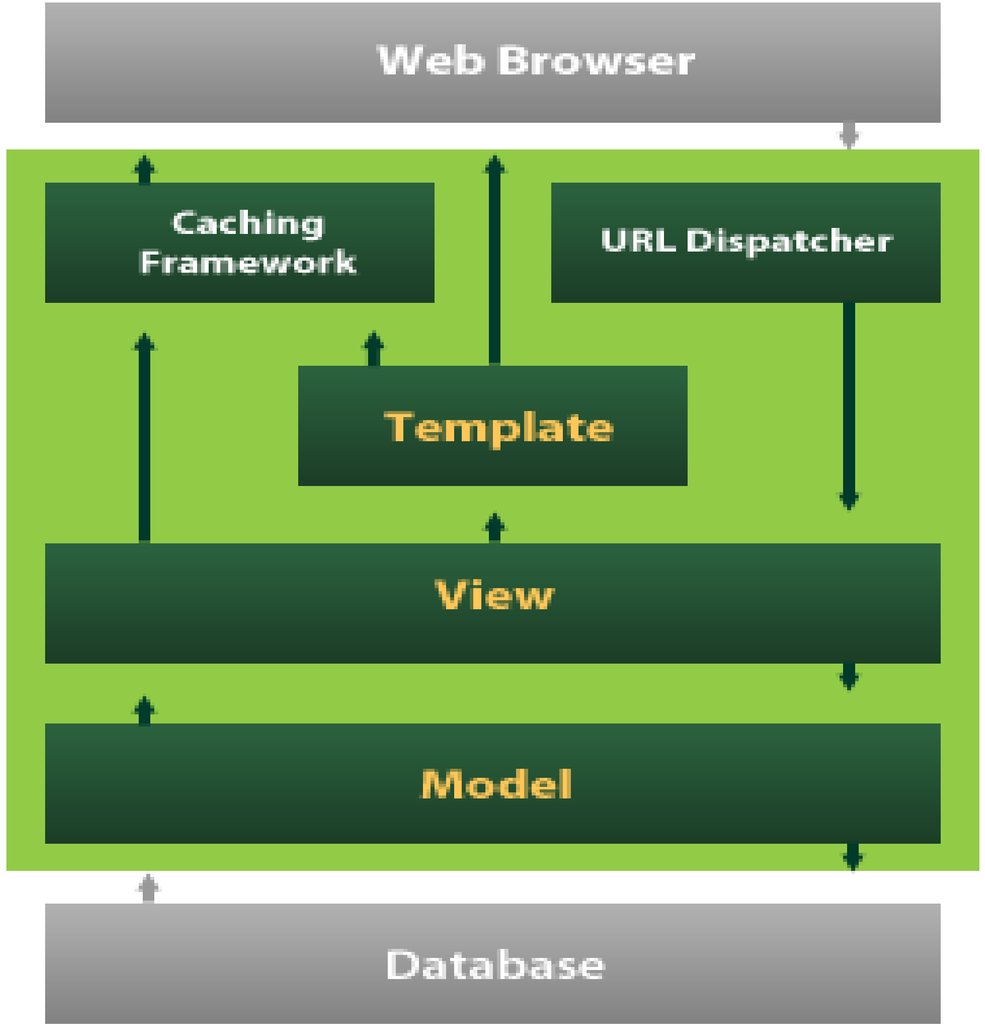


Django框架

- Django遵循MVC架构

  - M : 数据存取、处理模块，通过django orm实现。 (models.py)
  - V(T) : 对数据进行查询、逻辑处理，通过template展示。(views.py | templates\my_temp.html)
  - C : 根据用户输入委派视图的部分，由 Django 框架根据 URLconf 设置，对给定 URL 调用适当的 Python 函数。(urls.py)

Django框架


Web Browser
Caching Framework
URL Dispatcher
Template
View
Model
Database

Django框架

- Project VS Application

  - Application是一类web应用，它提供一些服务。
  - Project是这些web应用和配置的集合。
  - 一个web应用可以在多个project下面，同时一个project下面可以有多个web应用。

Django框架结构

Django-settings

- 获取配置项

```
from django.conf import settings
```

- 修改配置项MY_SETTING

```
# Test

> from django.conf import settings
> settings.MY_SETTING = 1
> settings.MY_SETTING
1
```

Django-settings

- INSTALLED_APPS

```
# ./settings.py
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.admin',
    # OTHER 3rd Party App
    'app_control',
    'account',
    'bkoauth',
)
INSTALLED_APPS += INSTALLED_APPS_CUSTOM

# config/settings_custom.py
INSTALLED_APPS_CUSTOM = (
    # add your app here...
    'home_application',
)
```

Django-settings

- Databases

```
# Test

> from django.db import connection
> cursor = connection.cursor()
> from django.db import connections
> cursor = connections['other_db'].cursor()
```

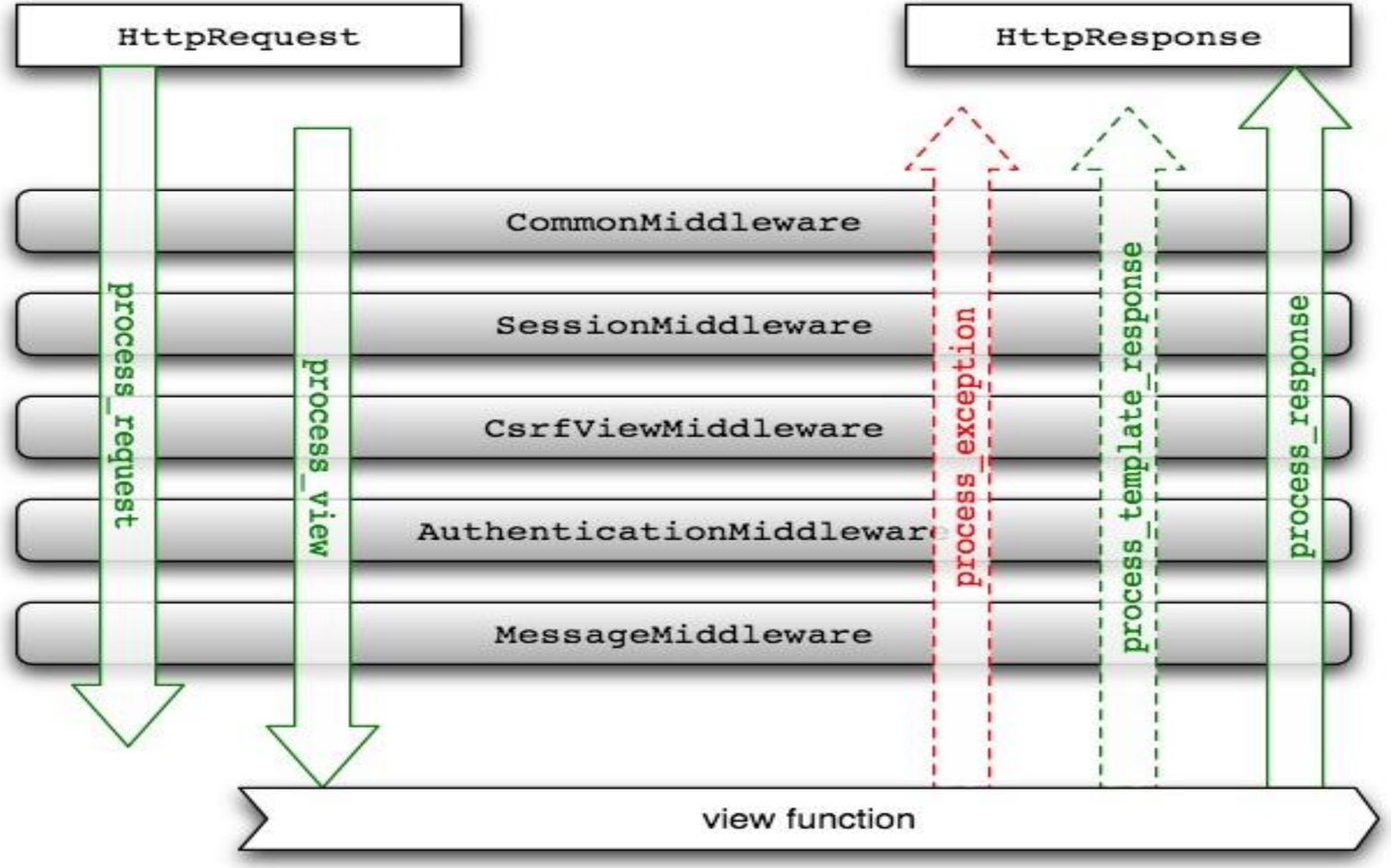
Django-settings[扩展]

- MIDDLEWARE_CLASSES

```
# ./settings.py
MIDDLEWARE_CLASSES = (
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    # 'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    # 'account.middlewares.LoginMiddleware',
    # 'common.middlewares.CheckXssMiddleware',
)
MIDDLEWARE_CLASSES += MIDDLEWARE_CLASSES_CUSTOM

# config/settings_custom.py
MIDDLEWARE_CLASSES_CUSTOM = (
)
```

Django-settings[扩展]

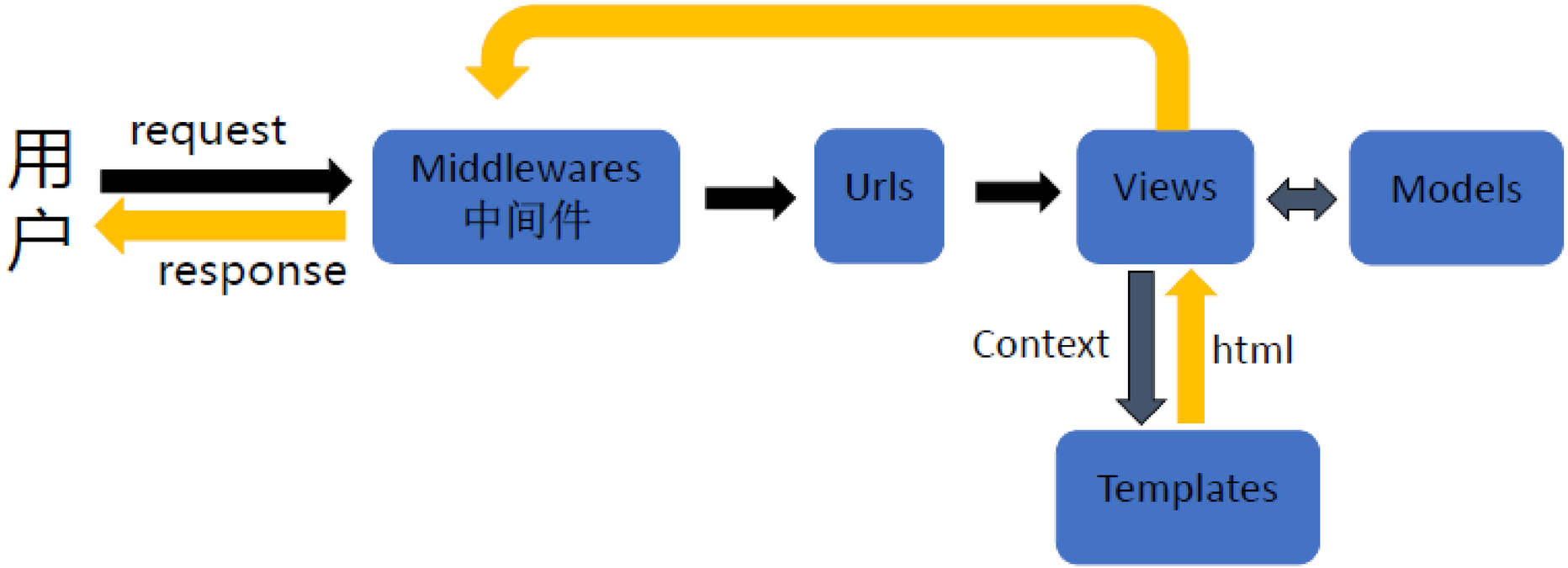


Django-URLConf & View

Django-URLConf & View

Concept：

To encapsulate the logic responsible for processing a user’s request and for returning the respo



Django-URLConf & View

- 项目配置APP入口

```
from django.conf.urls import patterns, include, url


# 用户自定义 urlconf
urlpatterns_custom = patterns(
    '',
    # 在home_application(根应用)里开始开发你的应用的主要功能
    # (此处home_application可以改成你想要的名字)
    url(r'^', include('home_application.urls')),
)
```

- APP 定制路由规则

```
from django.conf.urls import patterns

urlpatterns = patterns(
    'home_application.views',
    # 首页--your index
    (r'^$', 'home'),
    (r'^dev_guide/$', 'dev_guide'),
    (r'^contact/$', 'contact'),

    # For course
    (r'^mock_get/$', 'mock_get'),
    (r'^mock_post/$', 'mock_post')
)
```

Django-URLConf & View

- VIEW 示例（如何返回一个页面）

```python
from django.http import JsonResponse
from django.views.decorators.http import require_http_methods

from common.mymako import render_mako_context


def home(request):
    """
    首页
    """
    return render_mako_context(request, '/home_application/home.html')
```

Django-URLConf & View

- **VIEW 示例2**

```
urlpatterns = patterns(
    'home_application.views',
    # 首页--your index
    (r'^$', 'home'),
    (r'^dev_guide/$', 'dev_guide'),
    (r'^contact/$', 'contact'),

    (r'^instances/(?P<instance_id>[0-9]+)/update/$', 'update_instance'),
)
```

```
def update_instance(request, instance_id):
    """
    更新实例
    """
    return HttpResponse(instance_id)
```

Django-URLConf & View

- **HttpRequest对象（常用属性）**

  - HttpRequest.method
  - HttpRequest.GET
  - HttpRequest.POST
  - HttpRequest.COOKIES
  - HttpRequest.FILES
  - HttpRequest.session
  - HttpRequest.user
  - … …

Django-URLConf & View

- **HttpResponse对象** HttpResponse.__init__(*content=''*, *content_type=None*, *status=200*, *reason=None*, *charset=None*)

响应类型

- HttpResponse(**from** django.http **import** HttpResponse)
- JsonResponse (**from** django.http **import** JsonResponse)

返回页面

- render (**from** django.shortcuts **import** render)
- render_mako_context (**from** common.mymako **import** render_mako_context)

重定向

- HttpResponseRedirect (**from** django.http **import** HttpResponseRedirect)
- redirect (**from** django.shortcuts **import** redirect)
- reverse (**from** django.core.urlresolvers **import** reverse)

- ...

MoreInfo: https://docs.djangoproject.com/en/1.8/topics/http/shortcuts/

Django-URLConf & View

- **Request对象参数解析**

```
def organization_add(request):
    """
    创建组织
    """
    name = request.POST['name']
    modifier = request.user.username

    return JsonResponse({
        'result': True,
        'data': 'ok',
        'code': '200',
        'message': 'ok'
    })
```

Django-URLConf & View

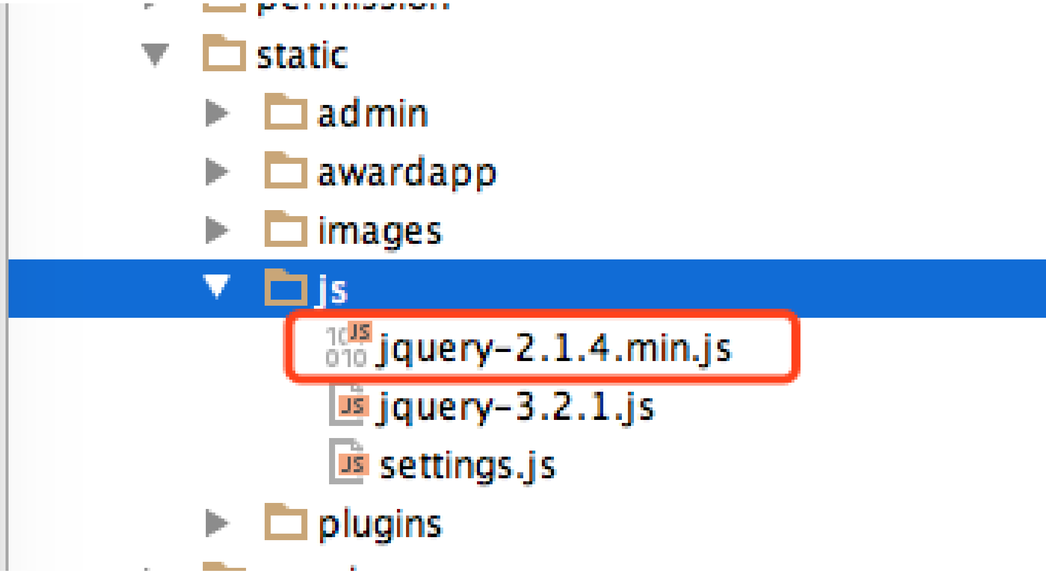
- 前端访问APP静态资源

```
<script src="{{ STATIC_URL }}js/jquery-2.1.4.min.js"></script>
```

```
{{ STATIC_URL }}

->

http://appdev.o.qcloud.com:8000/static/
```



奖项申报实战演示

Django-URLConf

- 奖项增删改查URL设计方案一

```
urlpatterns = patterns(
    'app1.views',
    url(r'^award/list/$', 'award_list', name='award'),
    url(r'^award/add/$', 'award_add', name='award_add'),
    url(r'^award/edit/(\d+)/$', 'award_edit', name='award_edit'),
    url(r'^award/del/(\d+)/$', 'award_del', name='award_del'),
    url(r'^award/detail/(\d+)/$', 'award_detail', name='award_detail'),
)
```

Django-Views

- 奖项列表

```
def award_list(request):
    """
    首页
    """
    ...
    context = {
        'award_list': award_list,
        'page_index': page_info.page_index
    }
    return render(request, 'awardapp/award_list.html', context)
```

Django-Views

- 奖项新增

```
1. def award_add(request):
2.     """
3.     奖项添加
4.     """
5.     if request.method == "GET":
6.         return render(request, 'awardapp/backstage/award_add.html', {})
7.     else:
8.         models.Award.objects.create(**request.POST)
9.         return HttpResponseRedirect(reverse('award_list'))
```

Django-Views

- 奖项编辑

```
1.  def award_edit(request, award_id):
2.      “””
3.      奖项编辑
4.      ”””
5.      context = {}
6.
7.      award_queryset = models.Award.objects.filter(id=award_id, is_delete=False)
8.
9.      if request.method == “GET”:
10.         context['award'] = award_queryset.first()
11.         return render(request, 'awardapp/award_edit.html', context)
12.     else:
13.         award_queryset.update(**request.POST)
14.
15.         return HttpResponseRedirect(reverse(‘award_detail’), args=(award_id,))
16.
```

Django-Views

- 奖项删除

```
1. def award_del(request, award_id):
2.     """
3.     奖项删除更新is_delete=True
4.     """
5.     models.Award.objects.filter(id=award_id).update(is_delete=True)
6.     return HttpResponseRedirect(reverse('award_list'))
```

Django-URLConf

- **RESTFUL API 设计原则**

GET	/api/countries/	Gets a list of countries
POST	/api/countries/	Gets a list of countries
GET	/api/countries/{pk}/	Detailed view of the country
PUT	/api/countries/{pk}/	Detailed view of the country
DELETE	/api/countries/{pk}/	Detailed view of the country
PATCH	/api/countries/{pk}/	Detailed view of the country

扩展学习 http://www.django-rest-framework.org/#

登录验证和用户信息获取

登陆验证和用户信息获取

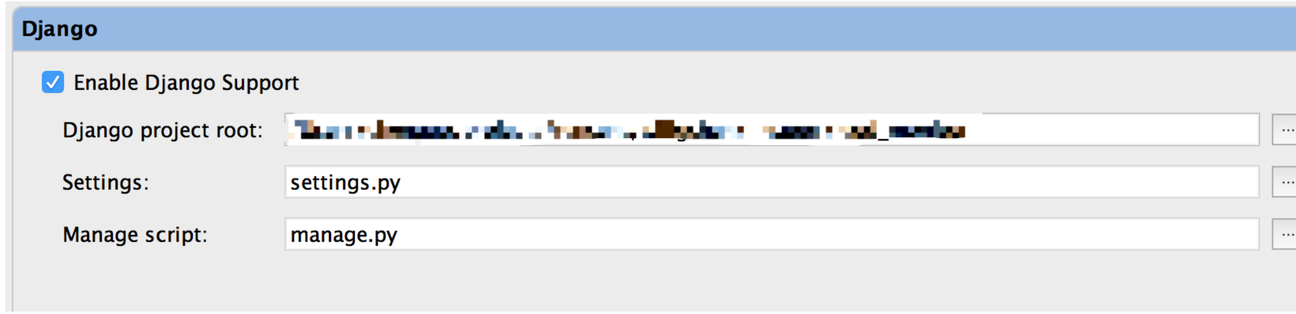
- **account模块**
  - 蓝鲸平台account模块使用了登陆验证的中间件，所有的请求都会通过登陆中间件的验证。
  - 需要不验证登陆态的请求，可以在views的函数上加一个装饰器@login_exempt用于豁免。
  - 登陆验证过程：从cookie中拿到open_id和open_key，其中open_id是一个用户的唯一身份，类似于账号。

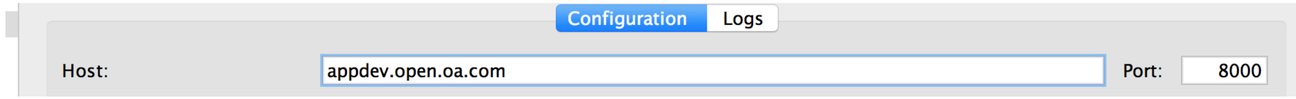
- **用户信息获取**
  - request.session[‘openid’] 或者 request.user.username
  - request.session[‘nick’] 昵称
  - request.session[‘avatar’] 头像url

Pycharm调试

Pycharm调试

- **Pycharm项目配置**
  - Django：
  - Project Structure：
- **调试配置**
  - Run->Edit Configurations->+Django Server
  - 配置本地域名（修改hosts指向localhost）
- **断点调试**
  - 运行debug
  - 在代码前面点击创建断点（小红点）
  - 访问页面，即可进入断点调试





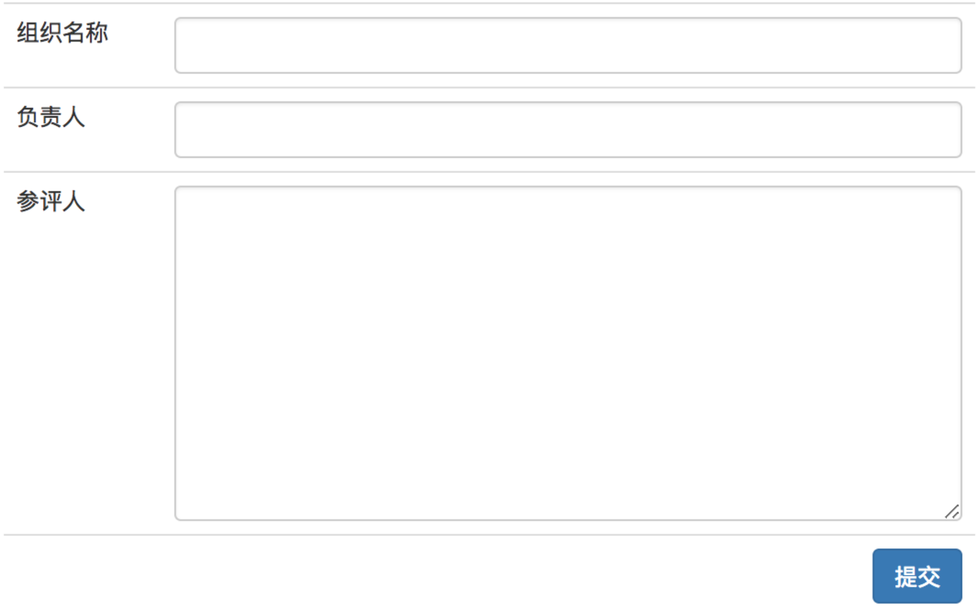
Django基础（一）课后作业

课后作业

- 前端访问APP静态资源
- 前端访问APP HTML页面
- urls文件配置项目路由规则

课后作业

- 后端解析表单提交



下集预告

Django基础（二）

- 文件上传下载
- Model 创建、字段说明、migrate 演示
- 简单 CURD 操作
- Admin 后台管理功能
- 实例演示创建数据


蓝鲸智云


腾讯云