ccorz

大音希声，小逼无形

博客园 首页 新随笔 联系 订阅 管理 随笔-161 文章-0 评论-16

公告

昵称：ccorz
园龄：2年3个月
粉丝：76
关注：12
+加关注

< 2018年8月 >

日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

python知识点(54)
python之web框架(24)
前端知识(16)
Django之奇技淫巧(13)
Tornado(4)
python(3)
Django RESTful framework(3)
python之数据库操作(3)
python之socket(2)
python之缓存(2)
更多

django-rest-framework 使用例子

Start Your API

创建项目

```
startproject rest_api
```

创建APP

```
startapp task
```

配置 rest_api/settings.py

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'task',

)
```

配置 rest_api/urls.py

```
# -*- coding:utf-8 -*-
from django.conf.urls import url, include

urlpatterns = [
    url(r'^api/', include('task.urls')),
]
```

创建模型 task/models.py

```
# -*- coding:utf-8 -*-
from django.db import models


class Task(models.Model):
    title = models.CharField('标题', max_length=100)
    description = models.TextField('描述')
    completed = models.BooleanField('是否完成',
default=False)
    create_date = models.DateTimeField('创建时间',
auto_now_add=True)
```

随笔档案

- 2018年5月 (2)
- 2018年4月 (3)
- 2017年7月 (1)
- 2017年6月 (7)
- 2017年4月 (1)
- 2017年3月 (9)
- 2017年2月 (28)
- 2017年1月 (4)
- 2016年12月 (3)
- 2016年11月 (9)
- 2016年10月 (9)
- 2016年9月 (13)
- 2016年8月 (15)
- 2016年7月 (14)
- 2016年6月 (25)
- 2016年5月 (18)

相册

- able(51)
- able1

常用链接

- Django新手图文指导
- jQuery语法介绍
- 免费SSL证书
- 申请免费SSL证书

积分与排名

- 积分 - 135329
- 排名 - 2404

最新评论

1. Re:Python之queue模块以及生产消费者模型

对博主的代码我这里有两个疑问：1. 生产者消费者一直不停么；另外主线程没有对子线程进行join()，资源怎么释放呢2. 假设生产者已经不再生产了，而此时q.empty() == True的话，那么多个......

--generalibm

2. Re:教你如何在linux上装逼，shell中颜色的设置

方法很好，可是这位大佬，您的颜色有点花哨啊

```
    def __unicode__(self):
        return self.title
```

序列化 task/serializers.py

```
# -*- coding:utf-8 -*-
from rest_framework import serializers
from .models import Task


class TaskSerializer(serializers.ModelSerializer):
    class Meta:
        model = Task
        fields = ('id', 'title', 'description',
'completed', 'create_date')
```

视图 task/views.py

```
# -*- coding:utf-8 -*-
from rest_framework.decorators import api_view
from rest_framework import status
from rest_framework.response import Response
from rest_framework.views import APIView
from rest_framework import generics

from .models import Task
from .serializers import TaskSerializer


# 第一种方式：APIView
class TaskList(APIView):
    def get(self, request, format=None):
        tasks = Task.objects.all()
        serializer = TaskSerializer(tasks, many=True)
        return Response(serializer.data)

    def post(self, request, format=None):
        serializer = TaskSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)


# 第二种方式：通用视图 ListCreateAPIView
class TaskListCreate(generics.ListCreateAPIView):
    queryset = Task.objects.all()
    serializer_class = TaskSerializer


# 第三种方式：装饰器 api_view
@api_view(['GET', 'POST'])
def task_list(request):
    '''
    List all tasks, or create a new task.
```

--gaby_yan

3. Re:Python操作MySQL之SQLAlchemy

解释的比较透,学习了.

--MAX_T

4. Re:django中的多级评论

这样在后端实现的递归，会增加服务器的压力

--黄土地上的黑石头

5. Re:Django之model详解

Good

--者行孙某

阅读排行榜

1. python中os.popen, os.system()区别(20834)
2. Django权限系统auth模块详解(18373)
3. 教你如何在linux上装逼，shell中颜色的设置(12707)
4. Django中authenticate和login模块(10775)
5. Django自定义用户认证系统之自定义用户模型(10580)

评论排行榜

1. Day9作业:socket之FTP工具(3)
2. Python基础一(2)
3. 获取Django model中字段名,字段的verbose_name,字段类型(2)
4. Day1作业2：多层菜单查询(1)
5. 教你如何在linux上装逼，shell中颜色的设置(1)

推荐排行榜

1. Python操作MySQL之SQLAlchemy(2)
2. Python之Redis操作(2)
3. django中同通过getlist() 接收页面form的post数组(2)
4. 爬虫相关-scrapy框架介绍(2)
5. ansible常用模块(2)

```
    '''
    if request.method == 'GET':
        tasks = Task.objects.all()
        serializer = TaskSerializer(tasks, many=True)
        return Response(serializer.data)
    elif request.method == 'POST':
        serializer = TaskSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)


@api_view(['GET', 'PUT', 'DELETE'])
def task_detail(request, pk):
    try:
        task = Task.objects.get(pk=pk)
    except Task.DoesNotExist:
        return Response(status=status.HTTP_404_NOT_FOUND)

    if request.method == 'GET':
        serializer = TaskSerializer(task)
        return Response(serializer.data)
    elif request.method == 'PUT':
        serializer = TaskSerializer(task,
data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        else:
            return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)
    elif request.method == 'DELETE':
        task.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)
```

路由 task/urls.py

```
# -*- coding:utf-8 -*-
from django.conf.urls import  url
from . import views

urlpatterns = [
    url(r'^tasks/$', views.task_list, name='task_list'),
    # url(r'^tasks/$', views.TaskList.as_view(),
name='task_list'),
    # url(r'^tasks/$', views.TaskListCreate.as_view(),
name='task_list'),
    url(r'^tasks/(?P<pk>[0-9]+)$', views.task_detail,
name='task_detail'),
]
```

Command Line

```
curl http://localhost:8000/api/tasks/
curl -X POST http://localhost:8000/api/tasks/ -d
"title=hello world&description=a whole new world"
curl -X PUT http://localhost:8000/api/tasks/1 -d
"title=hello world&description=be nice"
curl -X DELETE http://localhost:8000/api/tasks/1
```

Permissions & Authentication

创建项目

```
startproject rest_api
```

创建 APP

```
startapp task
```

配置 rest_api/settings.py

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'task',

)
```

配置 rest_api/urls.py

```
# -*- coding:utf-8 -*-
from django.conf.urls import url, include

urlpatterns = [
    url(r'^api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
    url(r'^api/', include('task.urls')),

]
```

创建模型 task/models.py

```
# -*- coding:utf-8 -*-
from django.db import models


class Task(models.Model):
    owner = models.ForeignKey('auth.User',
related_name='tasks')
    title = models.CharField('标题', max_length=100)
    description = models.TextField('描述')
    completed = models.BooleanField('是否完成',
default=False)
```

```
    create_date = models.DateTimeField('创建时间', auto_now_add=True)

    def __unicode__(self):
        return self.title
```

序列化 task/serializers.py

```
# -*- coding:utf-8 -*-
from rest_framework import serializers
from .models import Task


class TaskSerializer(serializers.ModelSerializer):
    owner = serializers.ReadOnlyField(source='owner.username')
    class Meta:
        model = Task
        fields = ('id', 'title', 'description', 'completed', 'owner')
```

创建 task/permissions.py

```
# -*- coding:utf-8 -*-
from rest_framework.permissions import IsAuthenticatedOrReadOnly, SAFE_METHODS


class IsOwnerOrReadOnly(IsAuthenticatedOrReadOnly):
    def has_object_permission(self, request, view, obj):
        if request.method in SAFE_METHODS:
            return True
        return obj.owner == request.user
```

视图 task/views.py

```
# -*- coding:utf-8 -*-
from rest_framework.generics import ListCreateAPIView, RetrieveUpdateDestroyAPIView

from .models import Task
from .serializers import TaskSerializer
from .permissions import IsOwnerOrReadOnly


class TaskMixin(object):
    queryset = Task.objects.all()
    serializer_class = TaskSerializer
    permission_classes = (IsOwnerOrReadOnly,)


class TaskList(TaskMixin, ListCreateAPIView):
    def perform_create(self, serializer):
        serializer.save(owner=self.request.user)
```

```
class TaskDetail(TaskMixin, RetrieveUpdateDestroyAPIView):
    pass
```

路由 task/urls.py

```
# -*- coding:utf-8 -*-
from django.conf.urls import  url
from . import views


urlpatterns = [
    url(r'^tasks/$', views.TaskList.as_view(),
name='task_list'),
    url(r'^tasks/(?P<pk>[0-9]+)$',
views.TaskDetail.as_view(), name='task_detail'),

]
```

Command Line

```
curl http://localhost:8000/api/tasks/
curl -X POST http://localhost:8000/api/tasks/ -d
"title=basic&description=BasicAuthentication" -u
admin:password
curl -X PUT http://localhost:8000/api/tasks/1 -d
"title=basic&description=BasicAuthentication&completed=True
" -u admin:password
curl -X DELETE http://localhost:8000/api/tasks/1 -u
admin:password

curl -X POST http://localhost:8000/api/tasks/ -H "X-
CSRFToken: token" --cookie "sessionid=session;
csrftoken=token" -d
"title=session&description=SessionAuthentication"
# X-CSRFToken in headers just need be the same with
csrftoken in cookie.
```

上面是两种默认的认证方式（ `SessionAuthentication` 和 `BasicAuthentication` ），还有 `TokenAuthentication` 。

通过以上，未授权认证的用户只能查看，只有授权认证的用户才可以进行创建，删除，更新操作。

标签: Django RESTful framework

好文要顶 关注我 收藏该文

ccorz
关注 - 12
粉丝 - 76
+加关注

0 0

« 上一篇: Django REST framework 的快速入门教程
» 下一篇: ansible 学习笔记

posted @ 2017-03-02 16:51 ccorz 阅读(1933) 评论(0) 编辑 收藏