**Django REST Framework教程:快速入门**

作者：吴秀峰 时间:2018年9月7日 20:26 点击次数:10206

我们将创建一个简单的允许管理员用户查看和编辑系统中的用户和组的API。

**项目设置**

创建一个名为 tutorial 的新django项目，然后启动一个名为 quickstart 的新app。

# 创建项目目录mkdir tutorial

cd tutorial# 创建一个virtualenv来隔离我们本地的包依赖关系virtualenv env

source env/bin/activate  # 在Windows下使用 `env\Scripts\activate`# 在创建的虚拟环境中安装 Django 和 Django REST frameworkpip install django

pip install djangorestframework# 创建一个新项目和一个单个应用django-admin.py startproject tutorial .  # 注意结尾的'.'符号cd tutorial

django-admin.py startapp quickstart

cd ..

现在第一次同步你的数据库：

python manage.py migrate

我们还要创建一个名为 admin 的初始用户，密码为 password123。我们稍后将在该示例中验证该用户。

python manage.py createsuperuser

等你建立好一个数据库和初始用户，并准备好开始。打开应用程序的目录，我们就要开始编码了...

**Serializers**

首先我们要定义一些序列化程序。我们创建一个名为 tutorial/quickstart/serializers.py的文件，来用作我们的数据表示。

from django.contrib.auth.models import User, Group

from rest\_framework import serializers

class UserSerializer(serializers.HyperlinkedModelSerializer):

    class Meta:

        model = User

        fields = ('url', 'username', 'email', 'groups')

class GroupSerializer(serializers.HyperlinkedModelSerializer):

    class Meta:

        model = Group

        fields = ('url', 'name')

请注意，在这个例子中我们用到了超链接关系，使用 HyperlinkedModelSerializer。你还可以使用主键和各种其他关系，但超链接是好的RESTful设计。

**Views**

好了，我们接下来再写一些视图。打开 tutorial/quickstart/views.py 文件开始写代码了。

from django.contrib.auth.models import User, Group

from rest\_framework import viewsets

from tutorial.quickstart.serializers import UserSerializer, GroupSerializer

class UserViewSet(viewsets.ModelViewSet):

    """

    允许用户查看或编辑的API路径。

    """

    queryset = User.objects.all().order\_by('-date\_joined')

    serializer\_class = UserSerializer

class GroupViewSet(viewsets.ModelViewSet):

    """

    允许组查看或编辑的API路径。

    """

    queryset = Group.objects.all()

    serializer\_class = GroupSerializer

不再写多个视图，我们将所有常见行为分组写到叫 ViewSets 的类中。

如果我们需要，我们可以轻松地将这些细节分解为单个视图，但是使用viewsets可以使视图逻辑组织良好，并且非常简洁。

**URLs**

好的，现在我们在tutorial/urls.py中开始写连接API的URLs。

from django.conf.urls import url, include

from rest\_framework import routers

from tutorial.quickstart import views

router = routers.DefaultRouter()

router.register(r'users', views.UserViewSet)

router.register(r'groups', views.GroupViewSet)

# 使用自动URL路由连接我们的API。

# 另外，我们还包括支持浏览器浏览API的登录URL。

urlpatterns = [

    url(r'^', include(router.urls)),

    url(r'^api-auth/', include('rest\_framework.urls', namespace='rest\_framework'))

]

因为我们使用的是viewsets而不是views，所以我们可以通过简单地使用路由器类注册视图来自动生成API的URL conf。

再次，如果我们需要对API URL进行更多的控制，我们可以简单地将其拉出来使用常规基于类的视图，并明确地编写URL conf。

最后，我们将包括用于支持浏览器浏览的API的默认登录和注销视图。这是可选的，但如果您的API需要身份验证，并且你想要使用支持浏览器浏览的API，那么它们很有用。

**Settings**

我们也想设置一些全局设置。我们想打开分页，我们希望我们的API只能由管理员使用。设置模块都在 tutorial/settings.py 中。

INSTALLED\_APPS = (

    ...

    'rest\_framework',

)

REST\_FRAMEWORK = {

    'DEFAULT\_PERMISSION\_CLASSES': [

        'rest\_framework.permissions.IsAdminUser',

    ],

    'PAGE\_SIZE': 10

}

好了，我们完成了。

**测试我们的API**

我们现在可以测试我们构建的API。我们从命令行启动服务器。

python manage.py runserver

我们现在可以从命令行访问我们的API，使用诸如 curl ...

bash: curl -H 'Accept: application/json; indent=4' -u admin:password123 http://127.0.0.1:8000/users/

{

    "count": 2,

    "next": null,

    "previous": null,

    "results": [

        {

            "email": "admin@example.com",

            "groups": [],

            "url": "http://127.0.0.1:8000/users/1/",

            "username": "admin"

        },

        {

            "email": "tom@example.com",

            "groups": [                ],

            "url": "http://127.0.0.1:8000/users/2/",

            "username": "tom"

        }

    ]

}

或者使用 httpie，命令行工具...

bash: http -a admin:password123 http://127.0.0.1:8000/users/

HTTP/1.1 200 OK

...

{

    "count": 2,

    "next": null,

    "previous": null,

    "results": [

        {

            "email": "admin@example.com",

            "groups": [],

            "url": "http://localhost:8000/users/1/",

            "username": "paul"

        },

        {

            "email": "tom@example.com",

            "groups": [                ],

            "url": "http://127.0.0.1:8000/users/2/",

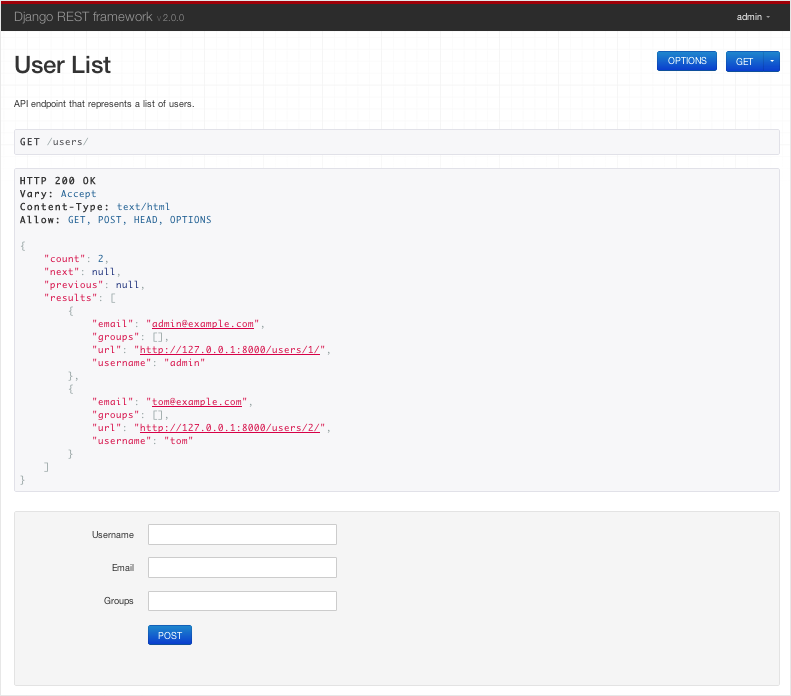
            "username": "tom"

        }

    ]

}

或直接通过浏览器，转到URL http://127.0.0.1:8000/users/...



如果您正在使用浏览器，请确保使用右上角进行登录。

非常棒！就是这么简单！