

HTML 与 Forms

DRF既可以返回API风格的响应，也可以返回普通的HTML页面，还可以将响应渲染为表单的形式。

渲染HTML页面

要返回HTML响应，你需要使用 `TemplateHTMLRenderer` 或者 `StaticHTMLRenderer` 渲染器

`TemplateHTMLRenderer` 需要一个上下文context数据，以及被渲染的html模板，类似Django的基本做法。

`StaticHTMLRender` 则需要一个预先已经渲染好的字符串格式的HTML内容。

下面是个例子：

views.py:

```
1  from my_project.example.models import Profile
2  from rest_framework.renderers import TemplateHTMLRenderer
3  from rest_framework.response import Response
4  from rest_framework.views import APIView
5
6
7  class ProfileList(APIView):
8      renderer_classes = [TemplateHTMLRenderer] # 指定渲染器
9      template_name = 'profile_list.html' # 指定模板
10
11     def get(self, request):
12         queryset = Profile.objects.all()
13         return Response({'profiles': queryset}) # 响应内容
```

profile_list.html:

```
1 <html><body>
2 <h1>Profiles</h1>
3 <ul>
4     {% for profile in profiles %}
5     <li>{{ profile.name }}</li>
6     {% endfor %}
7 </ul>
8 </body></html>
```

渲染表单

可以使用 `render_form` 这个DRF提供的模板标签，将序列化类的实例作为模板的上下文，把序列化类渲染为表单。

以下视图演示了在模板中使用序列化程序查看和更新模型实例的示例：

views.py:

```
1 from django.shortcuts import get_object_or_404
2 from my_project.example.models import Profile
3 from rest_framework.renderers import TemplateHTMLRenderer
4 from rest_framework.views import APIView
5
6
7 class ProfileDetail(APIView):
8     renderer_classes = [TemplateHTMLRenderer]
9     template_name = 'profile_detail.html'
10
11     def get(self, request, pk):
12         profile = get_object_or_404(Profile, pk=pk)
13         serializer = ProfileSerializer(profile)
14         return Response({'serializer': serializer, 'profile': profile})
15
16     def post(self, request, pk):
17         profile = get_object_or_404(Profile, pk=pk)
18         serializer = ProfileSerializer(profile, data=request.data)
19         if not serializer.is_valid():
20             return Response({'serializer': serializer, 'profile': profile})
21         serializer.save()
22         return redirect('profile-list')
```

profile_detail.html:

```
1  {% load rest_framework %}
2
3  <html><body>
4
5  <h1>Profile - {{ profile.name }}</h1>
6
7  <form action="{% url 'profile-detail' pk=profile.pk %}" method="POST">
8      {% csrf_token %}
9      {% render_form serializer %}
10     <input type="submit" value="Save">
11 </form>
12
13 </body></html>
```