

一、元数据Metadata

REST框架包含一个可配置的机制，用于确定API如何响应 `OPTIONS` 方式的HTTP请求。这允许你返回API模式或其他资源信息。

目前还没有任何广泛采用的约定来明确为 `OPTIONS` 请求返回哪种类型的响应，因此我们提供了一种特殊风格来返回一些有用的信息。

下面的示例演示了默认情况下响应的信息：

```
1  HTTP 200 OK
2  Allow: GET, POST, HEAD, OPTIONS
3  Content-Type: application/json
4
5  {
6      "name": "To Do List",
7      "description": "List existing 'To Do' items, or create a new item.",
8      "renders": [
9          "application/json",
10         "text/html"
11     ],
12     "parses": [
13         "application/json",
14         "application/x-www-form-urlencoded",
15         "multipart/form-data"
16     ],
17     "actions": {
18         "POST": {
19             "note": {
20                 "type": "string",
21                 "required": false,
22                 "read_only": false,
23                 "label": "title",
24                 "max_length": 100
25             }
26         }
27     }
28 }
```

可以通过 `'DEFAULT_METADATA_CLASS'` 设置全局性的元数据类：

```
1 REST_FRAMEWORK = {
2     'DEFAULT_METADATA_CLASS': 'rest_framework.metadata.SimpleMetadata'
3 }
```

或者通过 `metadata_class` 类属性，为每个类视图单独设置元数据:

```
1 class APIRoot(APIView):
2     metadata_class = APIRootMetadata
3
4     def get(self, request, format=None):
5         return Response({
6             ...
7         })
```

REST框架目前只有一个元数据类，也就是 `SimpleMetadata`。

如果您对创建使用常规的“get”请求访问的模式端点有特定的要求，那么您可能会考虑使用元数据API来实现这一点。

例如，可以在视图集中使用以下附加路由来提供可链接的模式端点。

```
1 @action(methods=['GET'], detail=False)
2 def schema(self, request):
3     meta = self.metadata_class()
4     data = meta.determine_metadata(request, self)
5     return Response(data)
```

二、自定义元数据

如果你想要自定义元数据类:

1. `BaseMetadata`
2. 实现 `determine_metadata(self, request, view)` 方法

下面的类可以限制给 `OPTIONS` 请求返回的信息:

```
1 class MinimalMetadata(BaseMetadata):
2     """
3     Don't include field and other information for `OPTIONS` requests.
4     Just return the name and description.
5     """
6     def determine_metadata(self, request, view):
7         return {
8             'name': view.get_view_name(),
9             'description': view.get_view_description()
10        }
```

自定义类后，如果想使用它，不要忘了在settings中进行配置：

```
1 REST_FRAMEWORK = {
2     'DEFAULT_METADATA_CLASS': 'myproject.apps.core.MinimalMetadata'
3 }
```