

一、内置的API文档功能

内置API文档包括：

- 每个API端点的文档。
- 为每个可用的API客户端自动生成代码样例
- 支持API交互

安装

依赖 `coreapi` 模块。 `pygments` 代码高亮和 `markdown` 编辑语法可选安装。

要使用文档，你还需要编写一条url：

```
1 from rest_framework.documentation import include_docs_urls
2
3 urlpatterns = [
4     ...
5     path('docs/', include_docs_urls(title='My API title'))
6 ]
```

这样会包含两个不同的视图

- `/docs/` - 文档页面本身
- `/docs/schema.js` - 一个js资源，暴露了API的概要。

为视图添加文档

例如：

```
1 class UserList(generics.ListAPIView):
2     """
3     Return a list of all the existing users.
4     """ # 也就是这些说明性的文字，不明白docstring是什么的，请学习python教程。
```

如果一个视图包含多个方法，那么你应该使用下面的格式，添加文档说明，以 `method:` 作为键：

```

1 class UserList(generics.ListCreateAPIView):
2     """
3     get: # 注意这里
4     Return a list of all the existing users.
5
6     post:
7     Create a new user instance.
8     """

```

使用ViewSet视图也是一样的:

```

1 class UserViewSet(viewsets.ModelViewSet):
2     """
3     retrieve:
4     Return the given user.
5
6     list:
7     Return a list of all the existing users.
8
9     create:
10    Create a new user instance.
11    """

```

自定义的方法也是一样的:

```

1 class UserViewSet(viewsets.ModelViewSet):
2     ...
3
4     @action(detail=False, methods=['get', 'post'])
5     def some_action(self, request, *args, **kwargs):
6         """
7         get:
8         A description of the get method on the custom action.
9
10        post:
11        A description of the post method on the custom action.
12        """
13
14        @some_action.mapping.put
15        def put_some_action():
16            """
17            A description of the put method on the custom action.
18            """

```

rest_framework.documentation 使用参考

`rest_framework.documentation` 模块提供了三个有用的函数帮助我们配置交互式的API文档。也就是 `include_docs_urls`、`get_docs_view` 和 `get_schemajs_view`。

include_docs_urls

- `title` : 默认为 `None` , 标题。
- `description` : 描述, 默认为 `None` 。
- `schema_url` : url路由, 默认为 `None` 。
- `public` : 默认为 `True` , 表示对外公开。
- `patterns` : url路由的列表, 默认为 `None`。
- `generator_class` : 文档生成器, 默认使用 `rest_framework.schemas.SchemaGenerator` 。
- `authentication_classes` : 认证模式, 默认为 `api_settings.DEFAULT_AUTHENTICATION_CLASSES` 。
- `permission_classes` : 权限验证类, 默认为 `api_settings.DEFAULT_PERMISSION_CLASSES` 。
- `renderer_classes` : 渲染器, 默认为 `None`。

get_docs_view

获取文档的视图, 具有以下参数, 意义基本同上。

- `title`
- `description`
- `schema_url`
- `public`
- `patterns`
- `generator_class`
- `authentication_classes`
- `permission_classes`
- `renderer_classes`

get_schemajs_view

参数也基本同上。

- `title`
- `description`
- `schema_url`
- `public`

- `patterns`
 - `generator_class`
 - `authentication_classes`
 - `permission_classes`
-

二、第三方文档模块

- drf-yasg - Yet Another Swagger Generator: <https://github.com/axnsan12/drf-yasg>
- DRF Docs
- Django REST Swagger : 已经被放弃
- DRF AutoDocs
- Apiary