

路径操作的高级配置

OpenAPI 的 operationId

!!! warning 如果你并非 OpenAPI 的「专家」，你可能不需要这部分内容。

你可以在路径操作中通过参数 `operation_id` 设置要使用的 OpenAPI `operationId`。

务必确保每个操作路径的 `operation_id` 都是唯一的。

```
{!../../../../../docs_src/path_operation_advanced_configuration/tutorial001.py!}
```

使用 路径操作函数 的函数名作为 operationId

如果你想用你的 API 的函数名作为 `operationId` 的名字，你可以遍历一遍 API 的函数名，然后使用他们的 `APIRoute.name` 重写每个 路径操作的 `operation_id`。

你应该在添加了所有 路径操作 之后执行此操作。

```
{!../../../../../docs_src/path_operation_advanced_configuration/tutorial002.py!}
```

!!! tip 如果你手动调用 `app.openapi()`，你应该在此之前更新 `operationId`。

!!! warning 如果你这样做，务必确保你的每个 路径操作函数 的名字唯一。

即使它们在不同的模块中（Python 文件）。

从 OpenAPI 中排除

使用参数 `include_in_schema` 并将其设置为 `False`，来从生成的 OpenAPI 方案中排除一个 路径操作（这样一来，就从自动化文档系统中排除掉了）。

```
{!../../../../../docs_src/path_operation_advanced_configuration/tutorial003.py!}
```

docstring 的高级描述

你可以限制 路径操作函数 的 `docstring` 中用于 OpenAPI 的行数。

添加一个 `\f`（一个「换页」的转义字符）可以使 **FastAPI** 在那一位置截断用于 OpenAPI 的输出。

剩余部分不会出现在文档中，但是其他工具（比如 Sphinx）可以使用剩余部分。

```
{!../../../../../docs_src/path_operation_advanced_configuration/tutorial004.py!}
```