直接返回响应

当你创建一个 **FastAPI** *路径操作* 时，你可以正常返回以下任意一种数据：`dict`，`list`，Pydantic 模型，数据库模型等等。

FastAPI 默认会使用 `jsonable_encoder` 将这些类型的返回值转换成 JSON 格式，`jsonable_encoder` 在 JSON 兼容编码器{.internal-link target=_blank} 中有阐述。

然后，**FastAPI** 会在后台将这些兼容 JSON 的数据（比如字典）放到一个 `JSONResponse` 中，该 `JSONResponse` 会用来发送响应给客户端。

但是你可以在你的 *路径操作* 中直接返回一个 `JSONResponse`。

直接返回响应可能会有用处，比如返回自定义的响应头和 cookies。

返回 `Response`

事实上，你可以返回任意 `Response` 或者任意 `Response` 的子类。

!!! tip "小贴士" `JSONResponse` 本身是一个 `Response` 的子类。

当你返回一个 `Response` 时，**FastAPI** 会直接传递它。

FastAPI 不会用 Pydantic 模型做任何数据转换，不会将响应内容转换成任何类型，等等。

这种特性给你极大的可扩展性。你可以返回任何数据类型，重写任何数据声明或者校验，等等。

在 `Response` 中使用 `jsonable_encoder`

由于 **FastAPI** 并未对你返回的 `Response` 做任何改变，你必须确保你已经准备好响应内容。

例如，如果不首先将 Pydantic 模型转换为 `dict`，并将所有数据类型（如 `datetime`、`UUID` 等）转换为兼容 JSON 的类型，则不能将其放入JSONResponse中。

对于这些情况，在将数据传递给响应之前，你可以使用 `jsonable_encoder` 来转换你的数据。

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

!!! note "技术细节" 你也可以使用 `from starlette.responses import JSONResponse`。

出于方便，`**FastAPI**` 会提供与 `` `starlette.responses` `` 相同的 `` `fastapi.responses` `` 给开发者。但是大多数可用的响应都直接来自 `Starlette`。

返回自定义 `Response`

上面的例子展示了需要的所有部分，但还不够实用，因为你本可以只是直接返回 `item`，而**FastAPI** 默认帮你把这个 `item` 放到 `JSONResponse` 中，又默认将其转换成了 `dict` 等等。

现在，让我们看看你如何才能返回一个自定义的响应。

假设你想要返回一个 XML 响应。

你可以把你的 XML 内容放到一个字符串中，放到一个 `Response` 中，然后返回。

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

说明

当你直接返回 `Response` 时，它的数据既没有校验，又不会进行转换（序列化），也不会自动生成文档。

但是你仍可以参考 OpenApI 中的额外响应{.internal-link target=_blank} 给响应编写文档。

在后续的章节中你可以了解到如何使用/声明这些自定义的 `Response` 的同时还保留自动化的数据转换和文档等。