

レスポンスを直接返す

FastAPI の *path operation* では、通常は任意のデータを返すことができます: 例えば、 `dict` 、 `list` 、Pydantic モデル、データベースモデルなどです。

デフォルトでは、**FastAPI** は [JSON互換エンコーダ](#){internal-link target=_blank} で説明されている `jsonable_encoder` により、返す値を自動的にJSONに変換します。

このとき背後では、JSON互換なデータ (例えば `dict`) を、クライアントへ送信されるレスポンスとして利用される `JSONResponse` の中に含めます。

しかし、*path operation* から `JSONResponse` を直接返すこともできます。

これは例えば、カスタムヘッダーやcookieを返すときに便利です。

Response を返す

実際は、 `Response` やそのサブクラスを返すことができます。

!!! tip "豆知識" `JSONResponse` それ自体は、 `Response` のサブクラスです。

`Response` を返した場合は、**FastAPI** は直接それを返します。

それは、Pydanticモデルのデータ変換や、コンテンツを任意の型に変換したりなどはしません。

これは多くの柔軟性を提供します。任意のデータ型を返したり、任意のデータ宣言やバリデーションをオーバーライドできます。

jsonable_encoder を Response の中で使う

FastAPI はあなたが返す `Response` に対して何も変更を加えないので、コンテンツが準備できていることを保証しなければなりません。

例えば、Pydanticモデルを `JSONResponse` に含めるには、すべてのデータ型 (`datetime` や `UUID` など) をJSON互換の型に変換された `dict` に変換しなければなりません。

このようなケースでは、レスポンスにデータを含める前に `jsonable_encoder` を使ってデータを変換できます。

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

!!! note "技術詳細" また、 `from starlette.responses import JSONResponse` も利用できます。

FastAPI は開発者の利便性のために `fastapi.responses` という `starlette.responses` と同じものを提供しています。しかし、利用可能なレスポンスのほとんどはStarletteから直接提供されます。

カスタム Response を返す

上記の例では必要な部分を全て示していますが、あまり便利ではありません。 `item` を直接返すことができるし、**FastAPI** はそれを `dict` に変換して `JSONResponse` に含めてくれるなど。すべて、デフォルトの動作です。

では、これを使ってカスタムレスポンスをどう返すか見てみましょう。

[XML](#)レスポンスを返したいとしましょう。

XMLを文字列にし、`Response` に含め、それを返します。

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

備考

`Response` を直接返す場合、バリデーションや、変換 (シリアライズ) や、自動ドキュメントは行われません。

しかし、[Additional Responses in OpenAPI](#){internal-link target=_blank}に記載されたようにドキュメントを書くこともできます。

後のセクションで、カスタム `Response` を使用・宣言しながら、自動的なデータ変換やドキュメンテーションを行う方法を説明します。