

カスタムレスポンス - HTML、ストリーム、ファイル、その他のレスポンス

デフォルトでは、**FastAPI** は `JSONResponse` を使ってレスポンスを返します。

[レスポンスを直接返す](#) {internal-link target=_blank} 見たように、`Response` を直接返すことでこの挙動をオーバーライドできます。

しかし、`Response` を直接返すと、データは自動的に変換されず、ドキュメントも自動生成されません (例えば、生成されるOpenAPIの一部としてHTTPヘッダー `Content-Type` に特定の「メディアタイプ」を含めるなど)。

しかし、*path operation* デコレータに、使いたい `Response` を宣言することもできます。

path operation 関数から返されるコンテンツは、その `Response` に含まれます。

そして、`Response` が、`JSONResponse` や `UJSONResponse` の場合のようにJSONメディアタイプ (`application/json`) ならば、データは *path operation* デコレータに宣言したPydantic `response_model` により自動的に変換 (もしくはフィルタ) されます。

!!! note "備考" メディアタイプを指定せずにレスポンスクラスを利用すると、FastAPIは何もコンテンツがないことを期待します。そのため、生成されるOpenAPIドキュメントにレスポンスフォーマットが記載されません。

ORJSONResponse を使う

例えば、パフォーマンスを出したい場合は、[orjson](#) をインストールし、`ORJSONResponse` をレスポンスとしてセットすることができます。

使いたい `Response` クラス (サブクラス) をインポートし、*path operation* デコレータに宣言します。

```
{!../../../docs_src/custom_response/tutorial001b.py!}
```

!!! info "情報" パラメータ `response_class` は、レスポンスの「メディアタイプ」を定義するために利用することもできます。

この場合、HTTPヘッダー `Content-Type` には `application/json` がセットされます。

そして、OpenAPIにはそのようにドキュメントされます。

!!! tip "豆知識" `ORJSONResponse` は、現在はFastAPIのみで利用可能で、Starletteでは利用できません。

HTMLレスポンス

FastAPI からHTMLを直接返す場合は、`HTMLResponse` を使います。

- `HTMLResponse` をインポートする。
- *path operation* のパラメータ `content_type` に `HTMLResponse` を渡す。

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

!!! info "情報" パラメータ `response_class` は、レスポンスの「メディアタイプ」を定義するために利用されます。

この場合、HTTPヘッダー `Content-Type` には `text/html` がセットされます。

そして、OpenAPIにはそのようにドキュメント化されます。

Response を返す

[レスポンスを直接返す](#){.internal-link target=_blank}で見たように、レスポンスを直接返すことで、*path operation* の中でレスポンスをオーバーライドできます。

上記と同じ例において、`HTMLResponse` を返すと、このようになります:

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

!!! warning "注意" *path operation*関数から直接返される `Response` は、OpenAPIにドキュメントされず (例えば、`Content-Type` がドキュメントされない)、自動的な対話的ドキュメントからも閲覧できません。

!!! info "情報" もちろん、実際の `Content-Type` ヘッダーやステータスコードなどは、返された `Response` オブジェクトに由来しています。

OpenAPIドキュメントと `Response` のオーバーライド

関数の中でレスポンスをオーバーライドしつつも、OpenAPIに「メディアタイプ」をドキュメント化したいなら、`response_class` パラメータを使い、`Response` オブジェクトを返します。

`response_class` はOpenAPIの *path operation* ドキュメントにのみ使用されますが、`Response` はそのまま使用されます。

`HTMLResponse` を直接返す

例えば、このようになります:

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

この例では、関数 `generate_html_response()` は、`str` のHTMLを返すのではなく `Response` を生成して返しています。

`generate_html_response()` を呼び出した結果を返すことにより、**FastAPI** の振る舞いを上書きする `Response` が既に返されています。

しかし、一方では `response_class` に `HTMLResponse` を渡しているため、**FastAPI** はOpenAPIや対話的ドキュメントでHTMLとして `text/html` でドキュメント化する方法を知っています。



利用可能なレスポンス

以下が利用可能なレスポンスの一部です。

`Response` を使って他の何かを返せまし、カスタムのサブクラスも作れることを覚えておいてください。

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

****FastAPI**** は開発者の利便性のために ``fastapi.responses`` として ``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。

Response

メインの `Response` クラスで、他の全てのレスポンスはこれを継承しています。

直接返すことができます。

以下のパラメータを受け付けます。

- `content` - `str` か `bytes` 。
- `status_code` - `int` のHTTPステータスコード。
- `headers` - 文字列の `dict` 。
- `media_type` - メディアタイプを示す `str` 。例えば `"text/html"` 。

FastAPI (実際にはStarlette) は自動的にContent-Lengthヘッダーを含みます。また、`media_type`に基づいたContent-Typeヘッダーを含み、テキストタイプのためにcharsetを追加します。

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

HTMLResponse

上で読んだように、テキストやバイトを受け取り、HTMLレスポンスを返します。

PlainTextResponse

テキストやバイトを受け取り、プレーンテキストのレスポンスを返します。

```
{!../../../../../docs_src/custom_response/tutorial005.py!}
```

JSONResponse

データを受け取り、 `application/json` としてエンコードされたレスポンスを返します。

上で読んだように、**FastAPI** のデフォルトのレスポンスとして利用されます。

ORJSONResponse

上で読んだように、[orjson](#) を使った、高速な代替のJSONレスポンスです。

UJSONResponse

[ujson](#) を使った、代替のJSONレスポンスです。

!!! warning "注意" `ujson` は、いくつかのエッジケースの取り扱いについて、Pythonにビルトインされた実装よりも作りこまれていません。

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

!!! tip "豆知識" `ORJSONResponse` のほうが高速な代替かもしれません。

`RedirectResponse`

HTTPリダイレクトを返します。デフォルトでは307ステータスコード (Temporary Redirect) となります。

```
{!../../../docs_src/custom_response/tutorial006.py!}
```

`StreamingResponse`

非同期なジェネレータか通常のジェネレータ・イテレータを受け取り、レスポンスボディをストリームします。

```
{!../../../docs_src/custom_response/tutorial007.py!}
```

`StreamingResponse` をファイルライクなオブジェクトとともに使う

ファイルライクなオブジェクト (例えば、`open()` で返されたオブジェクト) がある場合、`StreamingResponse` に含めて返すことができます。

これにはクラウドストレージとの連携や映像処理など、多くのライブラリが含まれています。

```
{!../../../docs_src/custom_response/tutorial008.py!}
```

!!! tip "豆知識" ここでは `async` や `await` をサポートしていない標準の `open()` を使っているため、通常の `def` で `path operation` を宣言していることに注意してください。

`FileResponse`

レスポンスとしてファイルを非同期的にストリームします。

他のレスポンスタイプとは異なる引数のセットを受け取りインスタンス化します。

- `path` - ストリームするファイルのファイルパス。
- `headers` - 含めたい任意のカスタムヘッダーの辞書。
- `media_type` - メディアタイプを示す文字列。セットされなかった場合は、ファイル名やパスからメディアタイプが推察されます。
- `filename` - セットされた場合、レスポンスの `Content-Disposition` に含まれます。

ファイルレスポンスには、適切な `Content-Length` 、 `Last-Modified` 、 `ETag` ヘッダーが含まれます。

```
{!../../../docs_src/custom_response/tutorial009.py!}
```

デフォルトレスポンスクラス

FastAPI クラスのインスタンスか `APIRouter` を生成するときに、デフォルトのレスポンスクラスを指定できます。

定義するためのパラメータは、`default_response_class` です。

以下の例では、**FastAPI** は、全ての *path operation* で `JSONResponse` の代わりに `ORJSONResponse` をデフォルトとして利用します。

```
{!../../../../../docs_src/custom_response/tutorial010.py!}
```

!!! tip "豆知識" 前に見たように、*path operation* の中で `response_class` をオーバーライドできます。

その他のドキュメント

また、OpenAPIでは `responses` を使ってメディアタイプやその他の詳細を宣言することもできます: [Additional Responses in OpenAPI](#){.internal-link target=_blank}