

Response Headers

Use a `Response` parameter

You can declare a parameter of type `Response` in your *path operation function* (as you can do for cookies).

And then you can set headers in that *temporal* response object.

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

And then you can return any object you need, as you normally would (a `dict`, a database model, etc).

And if you declared a `response_model`, it will still be used to filter and convert the object you returned.

FastAPI will use that *temporal* response to extract the headers (also cookies and status code), and will put them in the final response that contains the value you returned, filtered by any `response_model`.

You can also declare the `Response` parameter in dependencies, and set headers (and cookies) in them.

Return a `Response` directly

You can also add headers when you return a `Response` directly.

Create a response as described in [Return a Response Directly](#){internal-link target=_blank} and pass the headers as an additional parameter:

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

!!! note "Technical Details" You could also use `from starlette.responses import Response` or `from starlette.responses import JSONResponse`.

FastAPI provides the same `starlette.responses` as `fastapi.responses` just as a convenience for you, the developer. But most of the available responses come directly from Starlette.

And as the `Response` can be used frequently to set headers and cookies, **FastAPI** also provides it at `fastapi.Response`.

Custom Headers

Have in mind that custom proprietary headers can be added [using the 'X-' prefix](#).

But if you have custom headers that you want a client in a browser to be able to see, you need to add them to your CORS configurations (read more in [CORS \(Cross-Origin Resource Sharing\)](#){internal-link target=_blank}), using the parameter `expose_headers` documented in [Starlette's CORS docs](#).