

Response Cookies

Use a Response parameter

You can declare a parameter of type `Response` in your *path operation function*.

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

```
Python hl_lines="1 8-9" {!../../../../docs_src/response_cookies/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 cookies (also headers 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 cookies (and headers) in them.

Return a Response directly

You can also create cookies when returning a `Response` directly in your code.

To do that, you can create a response as described in Return a Response Directly.

Then set Cookies in it, and then return it:

```
Python hl_lines="10-12" {!../../../../docs_src/response_cookies/tutorial001.py!}
```

!!! tip Have in mind that if you return a response directly instead of using the `Response` parameter, FastAPI will return it directly.

So, you will have to make sure your data is of the correct type. E.g. it is compatible with

And also that you are not sending any data that should have been filtered by a ``response_model``

More info

!!! 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.

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

To see all the available parameters and options, check the documentation in Starlette.