

Cookie Parameters

You can define Cookie parameters the same way you define `Query` and `Path` parameters.

```
Import Cookie
```

First import `Cookie`:

Python 3.10+

```
from typing import Annotated
from fastapi import Cookie, FastAPI
app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[str | None, Cookie()] = None):
    return {"ads_id": ads_id}
```

► 📸 Other versions and variants

Python 3.9+

```
from typing import Annotated, Union
from fastapi import Cookie, FastAPI
app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[Union[str, None], Cookie()] = None):
    return {"ads_id": ads_id}
```

Python 3.8+

```
from typing import Union
from fastapi import Cookie, FastAPI
from typing_extensions import Annotated
app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[Union[str, None], Cookie()] = None):
    return {"ads_id": ads_id}
```

Python 3.10+ - non-Annotated

Tip

Prefer to use the `Annotated` version if possible.

```
from fastapi import Cookie, FastAPI
app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: str | None = Cookie(default=None)):
    return {"ads_id": ads_id}
```

Python 3.8+ - non-Annotated

Tip

Prefer to use the `Annotated` version if possible.

```
from typing import Union
from fastapi import Cookie, FastAPI
app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Union[str, None] = Cookie(default=None)):
```

```
    return {"ads_id": ads_id}
```

Declare Cookie parameters

Then declare the cookie parameters using the same structure as with `Path` and `Query`.

You can define the default value as well as all the extra validation or annotation parameters:

Python 3.10+

```
from typing import Annotated

from fastapi import Cookie, FastAPI

app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[str | None, Cookie()] = None):
    return {"ads_id": ads_id}
```

► 🌐 Other versions and variants

Python 3.9+

```
from typing import Annotated, Union

from fastapi import Cookie, FastAPI

app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[Union[str, None], Cookie()] = None):
    return {"ads_id": ads_id}
```

Python 3.8+

```
from typing import Union

from fastapi import Cookie, FastAPI
from typing_extensions import Annotated

app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Annotated[Union[str, None], Cookie()] = None):
    return {"ads_id": ads_id}
```

Python 3.10+ - non-Annotated

Tip

Prefer to use the `Annotated` version if possible.

```
from fastapi import Cookie, FastAPI

app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: str | None = Cookie(default=None)):
    return {"ads_id": ads_id}
```

Python 3.8+ - non-Annotated

Tip

Prefer to use the `Annotated` version if possible.

```
from typing import Union

from fastapi import Cookie, FastAPI

app = FastAPI()

@app.get("/items/")
async def read_items(ads_id: Union[str, None] = Cookie(default=None)):
```

```
return {"ads_id": ads_id}
```

Technical Details

`Cookie` is a "sister" class of `Path` and `Query`. It also inherits from the same common `Param` class.

But remember that when you import `Query`, `Path`, `Cookie` and others from `fastapi`, those are actually functions that return special classes.

Info

To declare cookies, you need to use `Cookie`, because otherwise the parameters would be interpreted as query parameters.

Recap

Declare cookies with `Cookie`, using the same common pattern as `Query` and `Path`.

© 2018 Sebastián Ramírez

Licensed under the MIT License.

<https://fastapi.tiangolo.com/tutorial/cookie-params/>

Exported from DevDocs — <https://devdocs.io>