

Global Dependencies

For some types of applications you might want to add dependencies to the whole application.

Similar to the way you can [add dependencies to the path operation decorators](#), you can add them to the `FastAPI` application.

In that case, they will be applied to all the *path operations* in the application:

Python 3.9+

```
from fastapi import Depends, FastAPI, Header, HTTPException
from typing_extensions import Annotated

async def verify_token(x_token: Annotated[str, Header()]):
    if x_token != "fake-super-secret-token":
        raise HTTPException(status_code=400, detail="X-Token header invalid")

async def verify_key(x_key: Annotated[str, Header()]):
    if x_key != "fake-super-secret-key":
        raise HTTPException(status_code=400, detail="X-Key header invalid")
    return x_key

app = FastAPI(dependencies=[Depends(verify_token), Depends(verify_key)])

@app.get("/items/")
async def read_items():
    return [{"item": "Portal Gun"}, {"item": "Plumbus"}]

@app.get("/users/")
async def read_users():
    return [{"username": "Rick"}, {"username": "Morty"}]
```

► Other versions and variants

Python 3.8+

```
from fastapi import Depends, FastAPI, Header, HTTPException
from typing_extensions import Annotated

async def verify_token(x_token: Annotated[str, Header()]):
    if x_token != "fake-super-secret-token":
        raise HTTPException(status_code=400, detail="X-Token header invalid")

async def verify_key(x_key: Annotated[str, Header()]):
    if x_key != "fake-super-secret-key":
        raise HTTPException(status_code=400, detail="X-Key header invalid")
    return x_key

app = FastAPI(dependencies=[Depends(verify_token), Depends(verify_key)])

@app.get("/items/")
async def read_items():
    return [{"item": "Portal Gun"}, {"item": "Plumbus"}]

@app.get("/users/")
async def read_users():
    return [{"username": "Rick"}, {"username": "Morty"}]
```

Python 3.8+ - non-Annotated

Tip

Prefer to use the `Annotated` version if possible.

```
from fastapi import Depends, FastAPI, Header, HTTPException

async def verify_token(x_token: str = Header()):
    if x_token != "fake-super-secret-token":
        raise HTTPException(status_code=400, detail="X-Token header invalid")

async def verify_key(x_key: str = Header()):
    if x_key != "fake-super-secret-key":
        raise HTTPException(status_code=400, detail="X-Key header invalid")
    return x_key

app = FastAPI(dependencies=[Depends(verify_token), Depends(verify_key)])
```

```
@app.get("/items/")
async def read_items():
    return [{"item": "Portal Gun"}, {"item": "Plumbus"}]

@app.get("/users/")
async def read_users():
    return [{"username": "Rick"}, {"username": "Morty"}]
```

And all the ideas in the section about [adding dependencies to the *path operation decorators*](#) still apply, but in this case, to all of the *path operations* in the app.

Dependencies for groups of *path operations*

Later, when reading about how to structure bigger applications ([Bigger Applications - Multiple Files](#)), possibly with multiple files, you will learn how to declare a single `dependencies` parameter for a group of *path operations*.

© 2018 Sebastián Ramírez
Licensed under the MIT License.
<https://fastapi.tiangolo.com/tutorial/dependencies/global-dependencies/>

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