



API Router

**tip:**

- Add your middleware  
at `main.py`

- Pass the session in the profile endpoint at `webapp/routes.py`





```
from fastapi import APIRouter, Depends, Request
```

```
@webapp_router.get("/profile", dependencies=[Depends(protected_endpoint)])
```

```
def profile(request: Request):
```

```
    return templates.TemplateResponse(
```

```
        "profile.html",
```

```
        {
```

```
            "request": request,
```

```
            "userinfo": request.session['userinfo']
```

```
        }
```

```
    )
```



**main.py**

**webapp/routes.py**

```
from starlette.middleware.sessions import SessionMiddleware
from config import config

. . .

app.add_middleware(
    SessionMiddleware,
    secret_key=config['WEBAPP']['SECRET_KEY']
)
```

# APIRouter

- Add your middleware at `main.py`
- Pass the session in the profile endpoint at `webapp/routes.py`

tip:

`main.py`

```
from starlette.middleware.sessions import SessionMiddleware
from config import config

...

app.add_middleware(
    SessionMiddleware,
    secret_key=config['WEBAPP']['SECRET_KEY']
)
```

`webapp/routes.py`

```
from fastapi import APIRouter, Depends, Request

@webapp_router.get("/profile", dependencies=[Depends(protected_endpoint)])
def profile(request: Request):

    return templates.TemplateResponse(
        "profile.html",
        {
            "request": request,
            "userinfo": request.session['userinfo']
        }
    )
```

# Flask

## Using Flask( ) or Blueprint( )

- Flask has a `session` object
- Session object has a dictionary interface

# FastAPI

## Using FastAPI( ) or APIRouter( )

- Requires the use of `SessionMiddleware`
- The middleware needs to be added to the application
- Sessions are part of the Request object after the middleware is added
- Session object has a dictionary interface