

# Working with HTTP and FastAPI

---



**Reindert-Jan Ekker**

@rjekker   [www.codesensei.nl](http://www.codesensei.nl)



# Overview



**Custom response types**

**Serving a website**

- Dynamic HTML with Jinja
- Processing Form Data

**Status codes and error handling**

**Headers and cookies**

**Middleware**

**Organizing our code with APIRouter**



# Operation Parameters and HTTP

```
from fastapi import Form, Cookie, Headers, Request, Response

@app.get(...)
def my_operation(
    # Form Data
    input1: int = Form(...), input2: str = Form(...),

    # Cookies or Headers
    my_cookie: str|None = Cookie(None),
    user_agent: str | None = Header(None)

    # Request and Response
    request: Request, response: Response):

    response.set_cookie(key="cookie_name", value="yummie!")
    response.headers["X-Pluralsight-Course-Header"] = "fastapi-course"
    response.status_code = 418
```



# Operations in Multiple Files

## routes/cars.py

```
from fastapi import APIRouter

router = APIRouter(prefix="/api/cars")

@router.get
def get_cars():
    ...
```

## carsharing.py

```
from routes import cars

app = FastAPI(title="Car Sharing")
app.include_router(cars.router)
```

# Serving HTML

```
from starlette.responses import HTMLResponse

@router.get("/", response_class=HTMLResponse)
def home():
    return "<html><body><h1>Welcome!</h1></body></html>"
```



# Handling Exceptions

```
from starlette.responses import JSONResponse
```

```
@app.exception_handler(SomeExceptionType)
async def badtrip_exception_handler(request: Request, exc: SomeExceptionType):
    return JSONResponse(
        content={"message": "Something went wrong"},
    )
```



# Middleware

```
@app.middleware("http")
async def my_middleware(request: Request, call_next):
    # do something with request
    log_request(request)

    # let FastAPI handle request
    response = await call_next(request)

    # do something with response
    change_response(response)
    return response
```



# Summary



**Organizing our code with APIRouter**

**Custom response types**

**Serving a website**

**Status codes and error handling**

**Headers and cookies**

**Middleware**





# Up Next: Authentication

---

