

# FastAPI Cheat Sheet

By 4u70mata

## 1. Installation

FastAPI is a modern, fast (high-performance) web framework for building APIs with Python 3.6+ based on standard Python type hints. It's designed for building RESTful APIs quickly and efficiently. **Key Features:**

- **Fast:** Built on Starlette for high performance.
- **Easy:** Easy to use + automatic interactive API documentation.
- **Type Safety:** Python type hints for data validation + serialization.
- **Asynchronous:** Async support, ideal for I/O-bound applications.

## 1. Installation

```
pip install fastapi uvicorn
```

## 2. Basic Setup

```
from fastapi import FastAPI

app = FastAPI()
```

## 3. Running the App

```
uvicorn main:app --reload
```

## 4. Routing

```
@app.get("/items/{item_id}")
async def read_item(item_id: int):
    return {"item_id": item_id}
```

## 5. Request Body

```
from pydantic import BaseModel

class Item(BaseModel):
    name: str
    price: float
    is_offer: bool = None

@app.post("/items/")
async def create_item(item: Item):
    return item
```

## 6. Response Models

```
@app.get("/items/{item_id}",
         response_model=Item)
async def read_item(item_id: int):
    return {"name": "Item", \
            "price": 10.5, \
            "is_offer": True}
```

## 7. Error Handling

```
from fastapi import HTTPException

@app.get("/items/{item_id}")
async def read_item(item_id: int):
    if item_id < 0:
        raise HTTPException(status_code=400, \
                             detail="Invalid item ID") \
    return {"item_id": item_id}
```

## 8. Middleware

```
from fastapi.middleware.cors \
    import CORSMiddleware

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

## 9. Background Tasks

```
from fastapi import BackgroundTasks

def write_log(message: str):
    with open("log.txt", mode="a") as log:
        log.write(message + "\n")

@app.post("/send-notification/")
async def send_notification(background_tasks:
                             BackgroundTasks, email: str):
    background_tasks.add_task(write_log, email)
    return {"message": "Notification sent"}
```

## 10. Static Files

```
from fastapi.staticfiles import StaticFiles

app.mount("/static", \
          StaticFiles(directory="static")
          , name="static")
```

## 11. Testing

```
from fastapi.testclient import TestClient

client = TestClient(app)

def test_read_root():
    response = client.get("/")
    assert response.status_code == 200
    assert response.json() == {"field": "xyz"}
```