Part 3: Testing And Deploying FastAPI Applications

8

Testing FastAPI Applications

9

Deploying FastAPI Applications

Index

Other Books You May Enjoy

Preface

FastAPI is a fast and efficient web framework for building APIs with Python. This book is a thorough guide on how to build an application with the FastAPI framework.

It starts with the basics of the FastAPI framework and the other technologies used throughout this book. You'll then learn about the different aspects of the framework: the routing system, response modeling, error handling, and templating.

In this book, you will learn how to build fast, efficient, and scalable applications in **Python** with **FastAPI**. You will begin from a *Hello World* application to a complete API that uses a database, authentication, and templates. You will learn how to structure your application to enhance efficiency, readability, and scalability. By integrating with other libraries in your application, you will learn how to connect your application to both a **SQL database** and a **NoSQL database**, integrate templates, and build authentication. Toward the end of this book, you will learn how to write tests, containerize your application, build a continuous integration and delivery pipeline using GitHub actions, and deploy your application to multiple cloud services. All of these will be taught via a theoretical and practical approach.

By the end of this book, you'll be equipped with the necessary knowledge to build and deploy a robust web API using the FastAPI framework.

Who this book is for

The primary audience for this book is any Python developer interested in building web APIs. The ideal reader is familiar with the basics of the Python programming language.

What this book covers

Chapter 1, *Getting Started with FastAPI*, introduces the basics of FastAPI and other technologies used in the book. The chapter also details the steps involved in setting up your development environment for your FastAPI application.

Chapter 2, *Routing in FastAPI*, talks in detail about the process of creating endpoints using the routing system in FastAPI. The components of a routing system, including the request body and path parameters, are also discussed alongside validating them with pydantic classes.

Chapter 3, *Response Models and Error Handling*, introduces responses in FastAPI, response modeling, error handling, and status codes.

Chapter 4, *Templating in FastAPI*, discusses how templates can be used to create views and render responses from the API.

Chapter 5, *Structuring FastAPI Applications*, introduces structuring applications and also briefly introduces the application to be built in the next chapters.

Chapter 6, *Connecting to a Database*, discusses two classes of databases (SQL and NoSQL) and demonstrates how you can connect your FastAPI application to either of them. We'll cover how to connect to and use a SQL database using SQLModel, and how to work with MongoDB using an object-document mapper, Beanie.

Chapter 7, *Securing FastAPI Applications*, talks about what securing your application entails – authorization and authentication, implementing authentication, and restricting access to application endpoints.

Chapter 8, *Testing FastAPI Applications*, explains what testing is and how to test our API endpoints.

Chapter 9, *Deploying FastAPI Applications*, discusses the steps involved in deploying your FastAPI application.

To get the most out of this book

You will need the latest version of Python installed on your computer. You also need to be familiar with the Python programming language to get the most out of this book.

Software/hardware covered in the book	**Operating system requirements**
Python 3.10	Windows, macOS, or Linux
Git 2.36.0	Windows, macOS, or Linux

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at `https://github.com/PacktPublishing/Building-Python-Web-APIs-with-FastAPI`. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://packt.link/qqhpc`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "To switch back to the original `main` branch, we run `git checkout main`."

A block of code is set as follows:

```
from fastapi import FastAPI
from routes.user import user_router

import uvicorn
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
from pydantic import BaseModel
from typing import List

class Event(BaseModel):
    id: int
    title: str
    image: str
    description: str
    tags: List[str]
    location: str
```

Any command-line input or output is written as follows:

```
$ git add hello.txt
$ git commit -m "Initial commit"
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "As shown in the previous model diagram, each user will have an **Events** field, which is a list of the events they have ownership of."

> **Tips or Important Notes**
>
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.