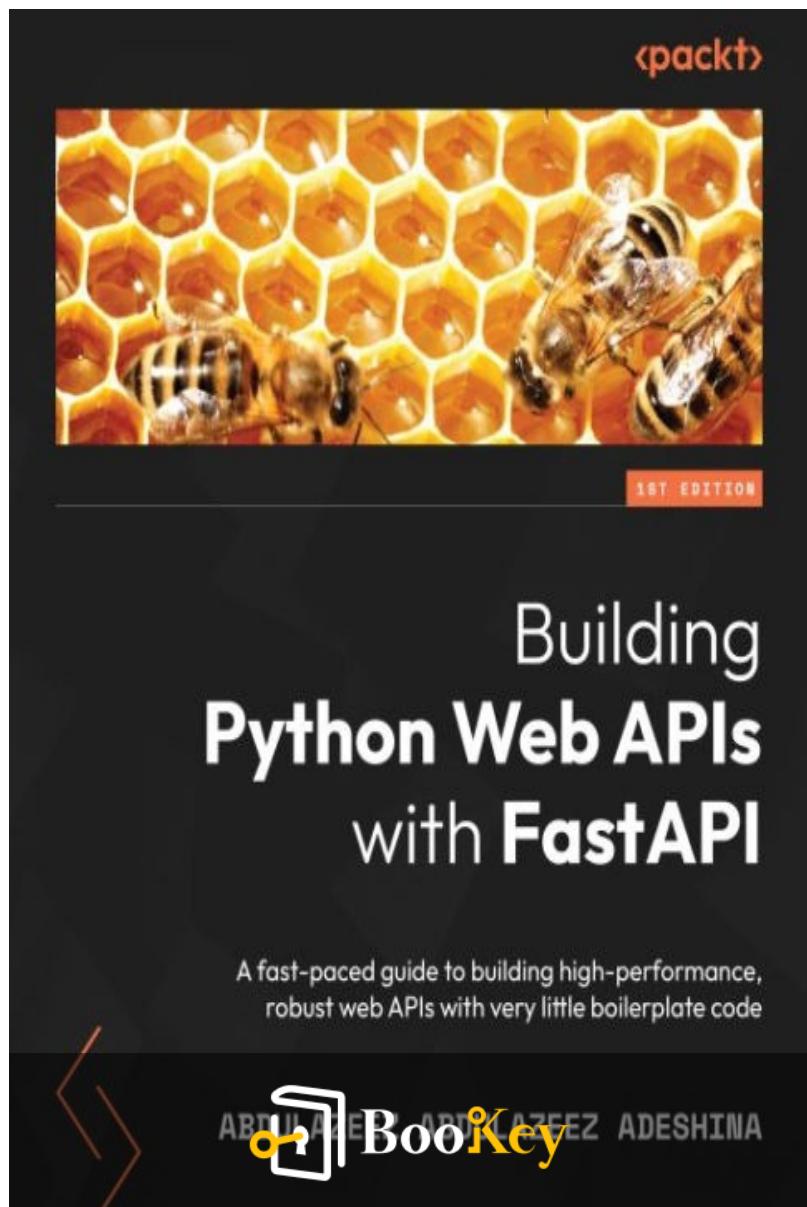


Building Python Web APIs With Fastapi PDF

Abdulazeez Abdulazeez Adeshina



More Free Books on Bookey



Scan to Download

Building Python Web Apis With Fastapi

Master FastAPI for Rapid, Robust API Development
in Python

Written by Bookey

[Check more about Building Python Web Apis With
Fastapi Summary](#)

[Listen Building Python Web Apis With Fastapi Audiobook](#)

More Free Books on Bookey



Scan to Download

About the book

Unlock the power of FastAPI and elevate your web API development skills with this comprehensive guide. **Building Python Web APIs with FastAPI** provides a hands-on approach to creating high-quality, production-ready APIs swiftly using Python. From the fundamentals of FastAPI's syntax and configuration to real-world implementations, this book covers everything you need to know. You'll gain practical insights on setting up a FastAPI application, managing user requests, and performing essential CRUD operations while seamlessly connecting to SQL and NoSQL databases. Additionally, it delves into error handling, authentication, application security, and deployment strategies for both local and cloud environments. Tailored for intermediate Python developers, especially those familiar with Django or Flask, this book is your gateway to mastering the FastAPI framework and building robust web APIs with ease and efficiency.

More Free Books on Bookey



Scan to Download

About the author

Abdulazeez Abdulazeez Adeshina is a dedicated software developer and educator with a strong focus on Python and its web development frameworks, particularly FastAPI. With a passion for teaching complex technical subjects in an accessible manner, Abdulazeez has cultivated a reputation for breaking down intricate concepts into digestible insights, making him a valuable resource for both beginners and experienced developers alike. His extensive experience in building robust web applications and APIs has equipped him with the practical knowledge needed to guide readers through the intricacies of FastAPI, showcasing its powerful capabilities for creating modern web services. Through his work, he aims to empower others to harness the full potential of Python programming and contribute to the evolving landscape of web development.

More Free Books on Bookey



Scan to Download

Ad



Scan to Download
Bookey



Try Bookey App to read 1000+ summary of world best books

Unlock 1000+ Titles, 80+ Topics

New titles added every week

Brand

Leadership & Collaboration

Time Management

Relationship & Communication



Business Strategy

Creativity

Public

Money & Investing

Know Yourself

Positive P

Entrepreneurship

World History

Parent-Child Communication

Self-care

Mind & Sp

Insights of world best books

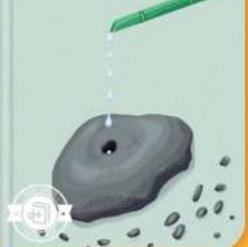
THINKING,
FAST AND SLOW
How we make decisions



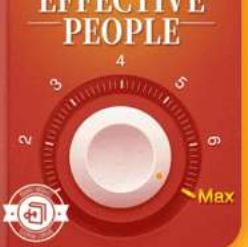
THE 48 LAWS OF POWER
Mastering the art of power, to have the strength to confront complicated situations



ATOMIC HABITS
Four steps to build good habits and break bad ones



THE 7 HABITS OF
HIGHLY
EFFECTIVE
PEOPLE



HOW TO TALK
TO ANYONE
Unlocking the Secrets of
Effective Communication



Free Trial with Bookey



Summary Content List

Chapter 1 : Getting Started with FastAPI

Chapter 2 : Routing in FastAPI

Chapter 3 : Response Models and Error Handling

Chapter 4 : Templating in FastAPI

Chapter 5 : Structuring FastAPI Applications

Chapter 6 : Connecting to a Database

Chapter 7 : Securing FastAPI Applications

Chapter 8 : Testing FastAPI Applications

Chapter 9 : Deploying FastAPI Applications

More Free Books on Bookey



Scan to Download

Chapter 1 Summary : Getting Started with FastAPI



Section	Content
Chapter 1	Getting Started with FastAPI
Overview	FastAPI is a modern, lightweight Python web framework for easy API creation, faster than Flask and Django, with a growing community.
Chapter Objectives	<ul style="list-style-type: none">Set up FastAPI environmentLearn Git basicsManage packages with pipCreate isolated environments using VirtualenvIntroduction to DockerBuild a "Hello World" FastAPI app
Key Topics	<ul style="list-style-type: none">Git BasicsSetting Up Virtual EnvironmentsPackage Management with pipIntroduction to DockerBuilding a FastAPI Application
Technical Requirements	Code files available on GitHub: GitHub Repository
Git Basics	<p>Installation: Download from git-scm.com.</p> <p>Common Commands: `git init`, `git commit`, `git status`.</p> <p>Branching: Create branches with `git checkout -b newbranch`.</p>

More Free Books on Bookey



Scan to Download

Section	Content
Creating Isolated Development Environments	<p>Create Virtual Environment: `mkdir todos && cd todos && python3 -m venv venv`.</p> <p>Activate & Deactivate: Use `source venv/bin/activate` to activate and `deactivate` to exit.</p>
Package Management with pip	<p>Installing Packages: `pip install fastapi`.</p> <p>Creating Requirements File: `pip freeze > requirements.txt`.</p>
Setting Up Docker	<p>Docker Setup: Download from Docker Installation Guide.</p> <p>Dockerfile: Define build instructions.</p> <p>Build and Run: Use `docker build` and `docker run` commands.</p>
Building a Simple FastAPI Application	<p>Activate the environment: `\\$ source venv/bin/activate`.</p> <p>Install Dependencies: `pip install fastapi uvicorn`.</p> <p>Create API: Define routes and handlers.</p> <p>Run Application: `uvicorn api:app --port 8000 --reload`.</p>
Conclusion	Introduced tools for FastAPI development; a simple app aids understanding, with upcoming topics on routing and building a CRUD app.

Chapter 1: Getting Started with FastAPI

Overview of FastAPI

FastAPI is a modern, lightweight Python web framework that

More Free Books on Bookey



Scan to Download

simplifies the creation of web APIs. Compared to other frameworks like Flask and Django, FastAPI is faster and easier to learn. It has a growing community and is popularly used for web APIs and machine learning model deployments.

Chapter Objectives

- Set up the development environment for FastAPI.
- Learn basics of version control with Git.
- Manage Python packages with pip.
- Create isolated development environments using Virtualenv.
- Get introduced to Docker.
- Build a simple "Hello World" FastAPI application.

Key Topics Covered

- Git Basics
- Setting Up Virtual Environments
- Package Management with pip
- Introduction to Docker
- Building a FastAPI Application

Technical Requirements

More Free Books on Bookey



Scan to Download

Code files for this chapter can be found on GitHub: [GitHub Repository](<https://github.com/PacktPublishing/Building-Python-Web-APIs-with-FastAPI/tree/main/ch01>)

Git Basics

Git is a decentralized version control system. Key operations:

- **Installation:**

Download Git from [git-scm.com](<https://git-scm.com/downloads>).

- **Common Commands:**

Initialize a repository with `git init`, commit changes with `git commit`, and track file status using `git status`.

- **Branching:**

Create branches to work on distinct features with `git checkout -b newbranch`.

Creating Isolated Development Environments with Virtualenv

More Free Books on Bookey



Scan to Download

Virtualenv is used to create isolated Python environments, preventing conflicts from globally installed packages.

-
- Creating a Virtual Environment:**

Run `mkdir todos && cd todos && python3 -m venv venv`.

-
- Activating & Deactivating:**

Use `source venv/bin/activate` to activate and `deactivate` to exit.

Package Management with pip

Pip simplifies the management of Python packages.

-
- Installing Packages:**

`pip install fastapi`.

-
- Creating Requirements File:**

Use `pip freeze > requirements.txt` to record installed packages.

Setting Up Docker

Docker is used for containerizing applications.

More Free Books on Bookey



Scan to Download

Docker Setup:

Download from [Docker Installation Guide](<https://docs.docker.com/install>).

Dockerfile:

Define application build instructions in a Dockerfile.

Build and Run Docker Containers:

Use commands like `docker build` and `docker run`.

Building a Simple FastAPI Application

To create a simple FastAPI app:

1.

Activate the environment:

`\$ source venv/bin/activate`.

2.

Install Dependencies:

Run `pip install fastapi uvicorn`.

3.

Create API:

Define routes and handlers using FastAPI.

4.

More Free Books on Bookey



Scan to Download

Run Application:

Use `uvicorn api:app --port 8000 --reload`.

Conclusion

This chapter introduced the tools required to start developing with FastAPI and provided a simple application to solidify understanding. In the next chapter, more on routing in FastAPI and building a CRUD application will be covered.

More Free Books on Bookey



Scan to Download

Example

Key Point:Importance of Setting Up a Proper Development Environment

Example:Imagine you're about to launch a project that requires clean organization and dependency management. As you dive into developing your FastAPI application, setting up a virtual environment becomes crucial; it ensures that all the packages you install are contained in a separate space, free from system-wide conflicts. You activate this environment with a simple command, feeling the assurance that your project is neatly packaged away from other projects. As you start to install FastAPI and its dependencies using pip, it's almost like laying down a solid foundation—everything runs seamlessly, and you can focus on building your application without worrying about version mismatches or package conflicts.

More Free Books on Bookey



Scan to Download

Chapter 2 Summary : Routing in FastAPI



Section	Content
Chapter Title	Routing in FastAPI
Key Topics Covered	<p>Understanding routing in FastAPI The APIRouter class for organizing routes Validation using Pydantic models Working with path and query parameters Building a simple CRUD application</p>
Understanding Routing	Connects HTTP requests to server-defined routes with specific handlers. FastAPI supports methods like GET, POST, PUT, PATCH, and DELETE.
APIRouter Class	Organizes multiple routes for better code maintenance and readability, allowing each route to be defined through its dedicated router instance.
Todo Application	Create routes for adding todos (POST) and retrieving them (GET) using APIRouter.
Include Routes	Use include_router() method to make APIRouter routes visible in the main FastAPI application.
Request Body Validation	Utilizes Pydantic models for validating input data to ensure data integrity and minimize security risks.
Path and Query Parameters	Path parameters identify resources; query parameters filter results based on criteria.
Request Body Usage	Data can be sent via POST or PUT methods. FastAPI uses Body() for validating request body structure.
Automatic Documentation	FastAPI auto-generates API documentation via Swagger and ReDoc for testing and clear descriptions.
CRUD	Build a complete CRUD application to manage todo items.

More Free Books on Bookey



Scan to Download

Section	Content
Application	
Summary	Focus on routing strategies, request validation, and CRUD application development, leading to further advanced topics.

Chapter 2: Routing in FastAPI

Routing is a fundamental aspect of constructing web applications, and FastAPI offers a flexible and straightforward approach to it. This chapter covers the basics of routing, including creating routes using the `APIRouter` class, utilizing models for request body validation, and understanding path and query parameters.

Key Topics Covered:

- Understanding routing in FastAPI
- The `APIRouter` class for organizing routes
- Validation using Pydantic models
- Working with path and query parameters
- Building a simple CRUD application

Understanding Routing in FastAPI

Routing connects HTTP requests from clients to

More Free Books on Bookey



Scan to Download

server-defined routes that have specific handlers for processing. A route handler is a function that executes when a request matches its route. FastAPI supports various HTTP methods like GET, POST, PUT, PATCH, and DELETE to perform actions on resources.

Routing Example

An initial routing setup can be done with a simple FastAPI instance. However, for applications with multiple routes, using the APIRouter class is more efficient.

Routing with the APIRouter Class

The APIRouter helps in organizing multiple routes within an application. This modularity enhances code maintenance and readability. Each route can be defined and managed through its dedicated router instance.

Creating a Todo Application

A simple Todo application is outlined in this chapter where two routes are created using APIRouter: one for adding todos (POST) and another for retrieving them (GET).

More Free Books on Bookey



Scan to Download

Including Routes in FastAPI Application

To make routes from the APIRouter visible in the main FastAPI application, the `include_router()` method is employed.

Validating Request Bodies with Pydantic Models

FastAPI allows for validating input data by utilizing Pydantic models, which are classes derived from the `BaseModel`. This validation ensures data integrity and minimizes security risks.

Using Path and Query Parameters

Path parameters, included in the URL, are used to identify resources, while query parameters help filter results based on specified criteria.

Request Body Usage

The request body can carry data sent to an API via methods such as `POST` or `PUT`. Proper validation of request bodies is

More Free Books on [Bookey](#)



Scan to Download

crucial, and FastAPI provides tools like Body() for ensuring correct data structure.

FastAPI Automatic Documentation

FastAPI auto-generates documentation for its APIs via Swagger and ReDoc, which enables testing and provides clear descriptions of API endpoints.

Building a CRUD App

The chapter concludes with building a complete CRUD application where routes for creating, reading, updating, and deleting todo items are created and tested effectively.

Summary

This chapter emphasizes routing strategies, request validation, and the building of a functioning CRUD application, setting the foundation for more advanced topics in subsequent chapters. In the next chapter, the focus will shift to response models and error handling in FastAPI.

More Free Books on Bookey



Scan to Download

Example

Key Point: Utilizing the APIRouter for organized route management streamlines application development.

Example: Imagine you are creating a web application where users can manage their personal tasks. By utilizing the APIRouter class in FastAPI, you can neatly organize your application's routes. For instance, you might create one router for user authentication and another for task management, allowing you to keep your code modular and readable. As you define routes for adding, retrieving, updating, and deleting tasks, you notice that managing these routes becomes significantly easier, enabling you to focus on implementing features rather than wrestling with tangled route definitions.

More Free Books on Bookey



Scan to Download

Critical Thinking

Key Point: Routing with APIRouter enhances modularity and maintainability.

Critical Interpretation: While the author emphasizes the importance of using the APIRouter for modularity in FastAPI routing, it could be argued that this approach might not be necessary for simpler applications. The claim for modularity should be evaluated critically as small-scale projects may benefit from a straightforward routing setup without additional complexity. Developers should consider whether the added structure of APIRouter truly enhances their specific project needs, especially in terms of overhead and learning curve. Sources such as Martin Fowler's writings and discussions on software design may provide contrasting perspectives on simplicity versus modularity in project structures.

More Free Books on Bookey



Scan to Download

Chapter 3 Summary : Response Models and Error Handling

Chapter 3: Response Models and Error Handling

Overview

Chapter 3 focuses on response models and error handling in FastAPI, explaining their significance in API development. Response models, built on Pydantic, serve as templates for returning structured data, while effective error handling practices ensure adequate status codes and messages are provided to clients.

Understanding Responses in FastAPI

Responses are crucial in the interaction between an API and its clients, typically formatted in JSON or XML. They consist of a header (providing status and additional info) and a body (containing the requested data).

More Free Books on Bookey



Scan to Download

Status Codes

HTTP status codes categorize responses into:

- 1XX: Request received
- 2XX: Successful request
- 3XX: Redirection
- 4XX: Client errors
- 5XX: Server errors

Common codes include 200 (OK), 404 (Not Found), and 500 (Internal Server Error).

Building Response Models

Response models dictate the format of returned data. They can be created using Pydantic, which allows filtering and organizing data fields. For example, defining a model for to-do items helps return only relevant fields in responses.

Install Bookey App to Unlock Full Text and Audio

More Free Books on Bookey



Scan to Download



Scan to Download



Why Bookey is must have App for Book Lovers

30min Content



The deeper and clearer interpretation we provide, the better grasp of each title you have.



Text and Audio format

Absorb knowledge even in fragmented time.



Quiz

Check whether you have mastered what you just learned.



And more

Multiple Voices & fonts, Mind Map, Quotes, IdeaClips...

Free Trial with Bookey



Chapter 4 Summary : Templating in FastAPI

Chapter 4: Templating in FastAPI

In this chapter, readers learn the process of rendering API responses on web pages using templates powered by Jinja, a Python-based templating language. Templating allows the display of API data in various formats, serving as a frontend component in web applications. By the end of the chapter, readers will understand what templating is and how to utilize it within FastAPI.

Topics Covered

- Understanding Jinja
- Using Jinja2 Templates in FastAPI
- Technical Requirements

Understanding Jinja

Jinja is a templating engine that facilitates the rendering of

More Free Books on Bookey



Scan to Download

API responses. Key components include:

-

Variable Blocks

: Wrapped in `{{ }}`, used for displaying values.

-

Control Structures

: Inside `{{% %}}`, this block includes if statements and loops.

-

Comments

: Denoted by `{{# #}}`, these are not rendered in the output.

Filters

Filters manipulate data in Jinja templates. They are applied using a pipe symbol (`|`) and can accept arguments. Common filters include:

-

Default Filter

: Provides a fallback value if the variable is None.

-

Escape Filter

: Renders raw HTML.

-

Conversion Filters

More Free Books on Bookey



Scan to Download

: Change data types (e.g., `int`, `float`).

Join Filter

: Combines elements from a list into a string.

Length Filter

: Calculates the length of an object.

Using if Statements

Similar to Python's syntax, if statements in Jinja are employed within control blocks to execute conditional logic.

Loops

Jinja supports iteration over lists or objects using loops.

Special variables, like `loop.index`, can be utilized inside loops.

Macros

Macros are reusable functions in Jinja, allowing for cleaner code by encapsulating repetitive HTML responses.

More Free Books on Bookey



Scan to Download

Template Inheritance

This powerful feature promotes the DRY principle, enabling child templates to inherit properties from a base template.

Using Jinja Templates in FastAPI

To implement templating, users must install `Jinja2` and create a directory for templates. The chapter guides users on creating templates such as `home.html` and `todo.html`, configuring FastAPI to serve these templates, and setting up routes to handle todo items.

Conclusion

Chapter 4 serves as a comprehensive introduction to using `Jinja` templating in `FastAPI`, focusing on how to render dynamic HTML content based on API responses, ultimately empowering users to enhance their web application's user interface.

More Free Books on [Bookey](#)



Scan to Download

Chapter 5 Summary : Structuring FastAPI Applications

Chapter 5: Structuring FastAPI Applications

In this chapter, we explore how to structure FastAPI applications as we transition from a simple single-file to-do application to a more organized event planner application. Proper structuring enhances readability, speeds up development, and increases productivity.

Topics Covered

- Structuring application routes and models
- Implementing models for a planner API

Application Structure

We will create an event planner with the following structure:

planner/
main.py

More Free Books on Bookey



Scan to Download

```
database/  
    __init__.py  
    connection.py  
routes/  
    __init__.py  
    events.py  
    users.py  
models/  
    __init__.py  
    events.py  
    users.py  
...  
...
```

Creating the Application Directory

1. Create a folder named `planner`.
2. Inside `planner`, create `main.py`, and three folders: `database`, `routes`, and `models`.
3. In each folder, create an `__init__.py` file, followed by specific files for database configuration, routing, and modeling.

Building the Event Planner Application

More Free Books on Bookey



Scan to Download

The event planner allows registered users to create, update, and delete events. Each user and event is assigned a unique ID to avoid conflicts.

Note: Database and authentication implementation will be further discussed in Chapters 6 and 7.

Setting Up the Development Environment

1. Create and activate a virtual environment.
2. Install the FastAPI and related dependencies.
3. Save these dependencies in `requirements.txt`.

Implementing Models

Models outline data storage and representation:

1. Define the `Event` model with fields like `id`, `title`, `image`, `description`, `tags`, and `location`.
2. Define the `User` model, including user credentials and event ownership.
3. Create additional models for new users and user sign-in.

Implementing Routes

Routes define API endpoints for user and event operations.

More Free Books on Bookey



Scan to Download

1. The User Routes

include:

- Sign-up: Users can register.
- Sign-in: Users can log in.

2. The Event Routes

manage event data:

- Retrieve all events or specific ones by ID.
- Add or delete events.

Testing the Application

Upon implementing the user and event routes, tests confirm that sign-up, sign-in, event creation, retrieval, and deletion functions as intended. The API also features interactive documentation through Swagger.

Summary

In this chapter, we learned to structure a FastAPI application and implement its routes and models for an event planning service. The next chapter will cover connecting the application to SQL and NoSQL databases, enhancing its capabilities by integrating database functionalities.

More Free Books on Bookey



Scan to Download

Chapter 6 Summary : Connecting to a Database

Chapter 6: Connecting to a Database

In this chapter, the focus is on transitioning a FastAPI application from using an in-app database to utilizing a proper SQL or NoSQL database. The aim is to enable persistent data storage, where data is retained even after app restarts or crashes. The chapter covers the following key topics:

Overview of Database Integration

- Explanation of databases as data storehouses, with details on organizing data into tables consisting of fields (columns) and records (rows).
- Introduction to the SQLAlchemy for SQL database operations and Beanie for MongoDB interactions.

Technical Requirements

More Free Books on Bookey



Scan to Download

- Instructions for installing MongoDB, with the code repository referenced for the chapter's resources.

Setting Up SQLModel

- Installation of the SQLModel library is required.
- Step-by-step creation of an SQL table model class using SQLModel, including defining columns and primary keys.

CRUD Operations with SQLModel

- Explanation of how to handle data insertion into the SQL database using the `Session` class.
- Definition of methods such as `add()`, `commit()`, `get()`, and an overview of creating the database using `create_engine()` and `metadata.create_all()`.
- Usage of the new SQLModel changes to structure event data model classes and CRUD functions.

Install Bookey App to Unlock Full Text and Audio

More Free Books on Bookey



Scan to Download



App Store
Editors' Choice



22k 5 star review



Scan to Download

Positive feedback

Sara Scholz

tes after each book summary
erstanding but also make the
and engaging. Bookey has
ding for me.

Fantastic!!!



Masood El Toure

I'm amazed by the variety of books and languages
Bookey supports. It's not just an app, it's a gateway
to global knowledge. Plus, earning points for charity
is a big plus!

José Botín

ding habit
o's design
ual growth

Love it!



Bookey offers me time to go through the
important parts of a book. It also gives me enough
idea whether or not I should purchase the whole
book version or not! It is easy to use!

Wonnie Tappkx

Time saver!



Bookey is my go-to app for
summaries are concise, ins-
curred. It's like having acc-
right at my fingertips!

Awesome app!



Rahul Malviya

I love audiobooks but don't always have time to listen
to the entire book! bookey allows me to get a summary
of the highlights of the book I'm interested in!!! What a
great concept !!!highly recommended!

Beautiful App



Alex Walk

This app is a lifesaver for book lovers with
busy schedules. The summaries are spot
on, and the mind maps help reinforce what
I've learned. Highly recommend!

Free Trial with Bookey



Chapter 7 Summary : Securing FastAPI Applications

Chapter 7: Securing FastAPI Applications

In this chapter, we focus on adding security to the FastAPI application using JSON Web Token (JWT) to restrict access to certain functionalities to authenticated users only. The aim is to prevent unauthorized access and ensure that sensitive actions, such as adding events, are protected.

Authentication and Authorization Basics

- Authentication verifies user credentials, while authorization determines permissions for actions.
- The chapter covers the following key topics:
 - Authentication methods in FastAPI.
 - Securing the application with OAuth2 and JWT.
 - Protecting routes using dependency injection.

Technical Requirements

More Free Books on Bookey



Scan to Download

- A MongoDB database is required, with code available on GitHub for reference.

Authentication Methods in FastAPI

FastAPI supports several authentication methods:

1.

Basic HTTP Authentication

: Credentials are sent with an Authorization header.

2.

Cookies

: Used to store user data on the client side.

3.

Bearer Token Authentication

: Involves using security tokens (like JWT) for authorization.

In this chapter, we will primarily utilize bearer token authentication.

Dependency Injection in FastAPI

- Dependency Injection allows the separation of concerns by injecting functions that provide parameters needed during request handling.
- Dependencies in FastAPI are defined as functions or classes

More Free Books on Bookey



Scan to Download

and called at runtime.

Securing the Application with OAuth2 and JWT

- The OAuth2 password flow is used, requiring a username and password to retrieve a JWT.
- JWT is signed using a secret key to prevent tampering.
- The chapter details the creation of necessary files and functions for handling JWTs, including password hashing using bcrypt.

Hashing Passwords

Users' passwords must not be stored in plain text. The Passlib library is used for secure password hashing.

Creating and Verifying Access Tokens

The payload of a JWT includes the user ID and expiration time, which is encoded and signed for security purposes.

Handling User Authentication

A function is implemented to retrieve and verify the user

More Free Books on Bookey



Scan to Download

based on the token provided. This dependency can then be injected into routes that require authentication.

Updating Application Routes

Routes are updated to:

- Include authentication for creating, updating, and deleting events.
- Ensure that only the event creator can modify or delete their respective events.
- New route definitions ensure security during user actions.

Configuring CORS

CORS middleware is added to the application to manage cross-origin requests, allowing responses from specific domains or any domain as defined by the configuration.

Summary

In summary, this chapter walks through:

- Securing FastAPI applications using OAuth2 and JWT.
- Utilizing dependency injection for authentication.
- Implementing secure password handling and JWT

More Free Books on Bookey



Scan to Download

management.

- Configuring CORS to manage access from various origins. This foundational knowledge strengthens the security and reliability of FastAPI applications, preparing us for future topics in testing and deploying the applications.

More Free Books on Bookey



Scan to Download

Critical Thinking

Key Point: The importance of secure authentication methods

Critical Interpretation: The author emphasizes using JWT for securing FastAPI applications, which is critical for protecting sensitive functionalities from unauthorized access. However, readers should note that while JWTs provide convenience and efficiency in stateless authentication, they can also introduce security challenges, such as token theft or improper implementation, as analyzed in various cybersecurity research papers and frameworks like OWASP's guidelines on security best practices (OWASP, 2023). Thus, adopting these methods without caution may lead to vulnerabilities, reinforcing the necessity for developers to critically evaluate the security of their implementations.

More Free Books on Bookey



Scan to Download

Chapter 8 Summary : Testing FastAPI Applications

Chapter 8: Testing FastAPI Applications

In this chapter, we delve into the importance and methodology of testing FastAPI applications. After establishing a secure API in the previous chapter using OAuth and JWT, we explore how to ensure its functionality through testing, marking a key step in the development cycle.

Overview of Testing

Testing is essential for verifying that an application operates as expected and for identifying issues prior to deployment. This chapter aims to transition manual endpoint testing to automated testing practices.

Key Topics Covered:

- Unit testing with pytest
- Setting up the test environment

More Free Books on Bookey



Scan to Download

- Writing tests for REST API endpoints
- Test coverage

Technical Requirements

To follow along, a running MongoDB server is necessary, with setup guidelines provided in Chapter 6. The source code is available on GitHub.

Unit Testing with pytest

Unit testing checks individual components of an application, using pytest due to its concise syntax and efficiency. A guide on installing pytest and structuring test files is included, demonstrating how to implement basic arithmetic operation tests.

Fixtures in pytest

Fixtures simplify test setup by providing reusable code. They can return necessary data for tests, and their scope can be defined (session or module). An example shows how to define a fixture for managing event instances.

More Free Books on Bookey



Scan to Download

Setting Up the Test Environment

To test asynchronous APIs, we introduce `httpx` and install `pytest-asyncio`. This involves creating a configuration file and initializing a client in the tests' setup.

Writing Tests for REST API Endpoints

Tests are structured for authentication and CRUD operations. The chapter details the creation of specific test files for sign-up, sign-in, and CRUD routes, employing `async` functions and verifying responses against expected results.

Test Coverage

Coverage testing measures how much of the code is tested. Instructions for installing the coverage module and generating reports (both terminal and HTML) help assess the thoroughness of the tests performed.

Summary

In summary, by the chapter's conclusion, readers are equipped with the skills to effectively test FastAPI

More Free Books on [Bookey](#)



Scan to Download

applications, employing pytest and its fixtures. The chapter prepares developers for publishing applications, with the upcoming chapter focused on deployment strategies using Docker and docker-compose.

More Free Books on Bookey



Scan to Download

Chapter 9 Summary : Deploying FastAPI Applications

Chapter 9: Deploying FastAPI Applications

In this chapter, the focus is on deploying FastAPI applications locally using Docker and docker-compose. The chapter begins by summarizing previous learning from testing API endpoints and the preparation needed before deployment.

Preparing for Deployment

Before deployment, it is crucial to ensure:

- Application dependencies are updated in the requirements.txt file.
- Environment variables are correctly configured.

Managing Dependencies

The requirements.txt file must be maintained to reflect the current application's dependencies, which can be updated

More Free Books on Bookey



Scan to Download

using the `pip freeze` command. A sample requirements.txt is provided, listing essential packages.

Configuring Environment Variables

Environment variables, elaborated in Chapter 6, must be handled properly, ensuring they remain out of version control systems.

Deploying with Docker

Docker enables containerization, allowing applications to run consistently across environments. The chapter provides details on writing a Dockerfile specific for the FastAPI application, describing each instruction, including setting the base image, working directory, copying files, running commands, and exposing ports.

Install Bookey App to Unlock Full Text and Audio

More Free Books on Bookey



Scan to Download



Scan to Download



Read, Share, Empower

Finish Your Reading Challenge, Donate Books to African Children.

The Concept



×



×



This book donation activity is rolling out together with Books For Africa. We release this project because we share the same belief as BFA: For many children in Africa, the gift of books truly is a gift of hope.

The Rule



Earn 100 points

Redeem a book

Donate to Africa

Your learning not only brings knowledge but also allows you to earn points for charitable causes! For every 100 points you earn, a book will be donated to Africa.

Free Trial with Bookey



Best Quotes from Building Python Web Apis With Fastapi by Abdulazeez Abdulazeez Adeshina with Page Numbers

[View on Bookey Website and Generate Beautiful Quote Images](#)

Chapter 1 | Quotes From Pages 36-82

1. FastAPI is the Python web framework that we are going to use in this book.
2. FastAPI is relatively new, but it has a growing community.
3. An understanding of the technologies previously mentioned is required to build a full-blown FastAPI application.
4. To enable tracking of files, a file must first be added and committed.
5. Branches are an important feature that enables developers to easily work on different application features, bugs, and so on, separately before merging into the main branch.
6. A virtual environment is an isolated environment where application dependencies installed can only be accessed

More Free Books on Bookey



Scan to Download

within it.

- 7.Pip is a Python package manager like JavaScript's yarn; it enables you to automate the process of installing Python packages – both globally and locally.
- 8.Docker is an efficient tool for containerization.
- 9.Our aim in this section is to introduce FastAPI by building a simple application.

Chapter 2 | Quotes From Pages 83-163

- 1.Routing is the process of handling HTTP requests sent from a client to the server.
- 2.The knowledge of routing in FastAPI is essential in building small- and large-scale applications.
- 3.The APIRouter class encourages modularity and organization of application routing and logic.
- 4.Validation using Pydantic models ensures only defined data is sent.
- 5.A model in FastAPI is a structured class that dictates how data should be received or parsed.
- 6.FastAPI generates JSON schema definitions for our models

More Free Books on Bookey



Scan to Download

and automatically documents our routes.

- 7.By validating our request body, we were able to ensure that proper data is sent to the API.
- 8.The inclusion of path parameters to our routes also enabled us to retrieve and delete a single todo from our todo list.

Chapter 3 | Quotes From Pages 164-191

- 1.Responses are an integral part of an API's life cycle.
- 2.The standard practice followed in building web applications, irrespective of the framework, is to return appropriate status codes for individual events.
- 3.HTTPException allows us to return an adequate error response code.
- 4.We have learned how to return the right response codes to clients, as well as overriding the default status code, in this section.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1 Million+ Quotes

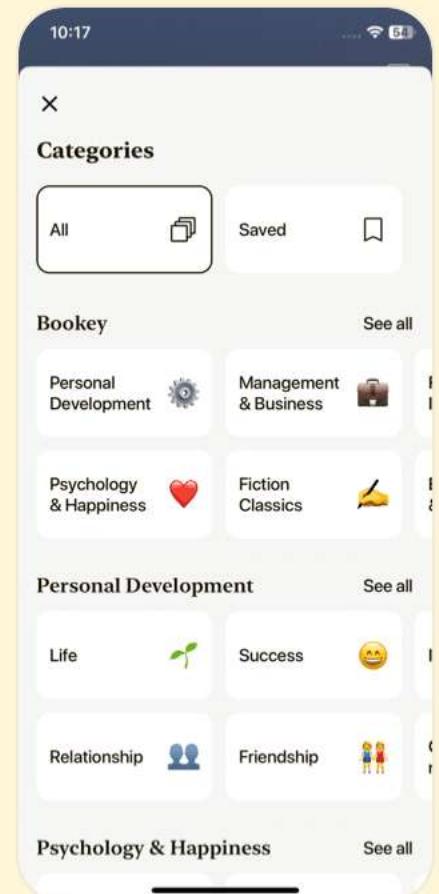
1000+ Book Summaries

Free Trial Available!

Scan to Download



Download on the
App Store



Chapter 4 | Quotes From Pages 192-220

1. The Jinja templating engine makes use of curly brackets { } to distinguish its expressions and syntax from regular HTML, text and any other variable in the template file.
2. Template inheritance is a situation where a base template is defined and child templates can interact, inherit, and replace defined sections of the base template.
3. Filters are an important part of every templating engine and in Jinja, filters enable us to execute certain functions such as joining values from a list and retrieving the length of an object, among others.
4. Now that we have learned what macros are, we will proceed to learn what template inheritance is and how it works in FastAPI.
5. Despite the similarity between Python and Jinja's syntax, modifications such as joining strings, setting the first character of a string to uppercase, and so on cannot be done using Python's syntax in Jinja.

More Free Books on Bookey



Scan to Download

6.By the end of this chapter, you will be equipped with the knowledge of what templating is and how to use templates to render information from your API.

Chapter 5 | Quotes From Pages 223-248

1.Structuring refers to the arrangement of application components in an organized format, which can be modular to improve the readability of the application's code and content.

2.An application with proper structuring enables faster development, faster debugging, and an overall increase in productivity.

3.Now that we have successfully structured our API and grouped similar files with respect to their functions into components, let's begin the implementation of the application in the next section.

4.Let's define the models for the event and user. The models describe how data will be stored, inputted, and represented in our application.

5.In the sign-up route defined previously, we're making use

More Free Books on Bookey



Scan to Download

of an in-app database. (We'll be introducing a database in Chapter 6, Connecting to a Database.)

6. We have successfully implemented the routes and models for our planner application. We have also tested them to assess their working status.

Chapter 6 | Quotes From Pages 249-385

1. By the end of this chapter, you will be equipped with the knowledge of how to connect a FastAPI application to a database.

2. A database can be simply referred to as a storehouse for data.

3. The Session class takes an argument that is the instance of a SQL engine.

4. The create_all() method creates the database as well as the tables defined.

5. A primary key is a unique identifier for a record contained in a database table.

6. We have successfully incorporated a SQL database into our application using SQLModel, as well as implementing

More Free Books on Bookey



Scan to Download

CRUD operations.

7. We are invoking the methods we defined in the database file earlier.

8. We'll take a look at proper authentication procedures in the next chapter.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1 Million+ Quotes

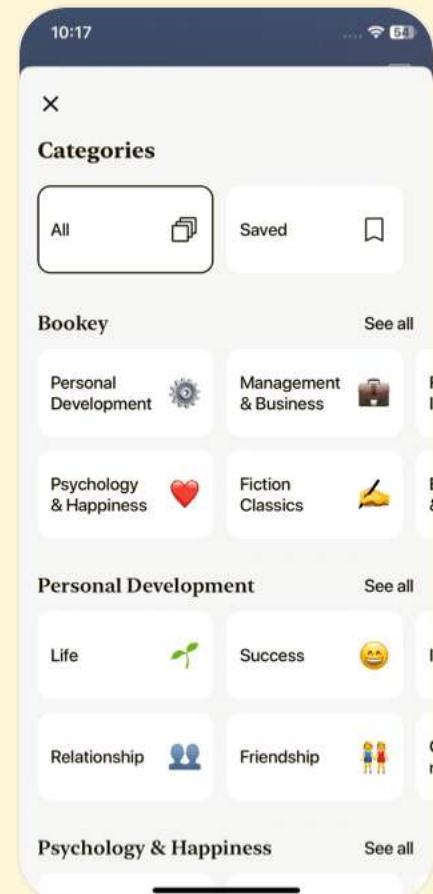
1000+ Book Summaries

Free Trial Available!

Scan to Download



Download on the
App Store



Chapter 7 | Quotes From Pages 386-464

1. Securing an application involves the addition of security measures to restrict access to application functionalities from unauthorized entities to prevent hacks or illegal modifications of the application.
2. Authentication is the process of verifying the credentials passed by an entity and authorization simply means giving an entity permission to perform designated actions.
3. By injecting the User model into the user function argument, we can easily retrieve the attributes of the object.
4. JWTs are signed using a unique key known only to the server and client to avoid the encoded string being tampered with by an external body.
5. In this chapter, we learned how to secure a FastAPI application with OAuth and JWT.

Chapter 8 | Quotes From Pages 467-536

1. Testing is an integral part of the application development cycle.

More Free Books on Bookey



Scan to Download

2. Unit testing is a testing procedure where individual components of an application are tested.
3. Fixtures are reusable functions defined to return the data needed in test functions.
4. A test coverage report is useful in determining the percentage of our code that was executed in the course of testing.
5. You have learned what testing is and how to write tests with pytest, a fast testing library built for Python applications.

Chapter 9 | Quotes From Pages 537-594

1. Deployment usually marks the end of an application's life cycle.
2. It is important that the requirements.txt file is kept up to date.
3. Environment variables are to be properly handled and kept out of version control systems such as GitHub.
4. Docker is the most popular technology used for containerization.

More Free Books on Bookey



Scan to Download

5. Each set of instructions listed in the Dockerfile is built as an individual layer.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1 Million+ Quotes

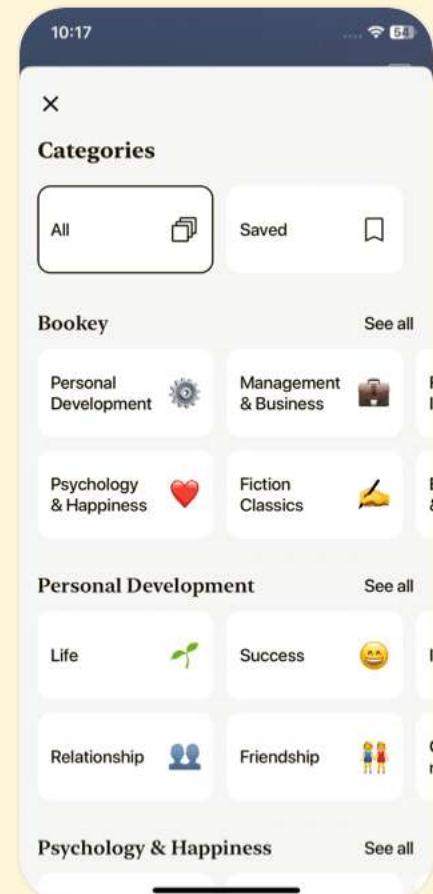
1000+ Book Summaries

Free Trial Available!

Scan to Download



Download on the
App Store



Building Python Web Apis With Fastapi Questions

[View on Bookey Website](#)

Chapter 1 | Getting Started with FastAPI| Q&A

1.Question

What is FastAPI, and why is it recommended for building web APIs?

Answer:FastAPI is a modern, fast, lightweight

Python web framework used for building APIs. It is recommended because it offers a simpler learning curve compared to other frameworks like Flask and Django, while also being efficient in performance, making it an excellent choice for developers.

2.Question

Why is it important to learn Git for developing FastAPI applications?

Answer:Learning Git is crucial because it provides version control for tracking changes in your project. As you build an application, Git helps manage revisions, making it easier to

More Free Books on Bookey



Scan to Download

collaborate on code and revert to previous versions if necessary.

3.Question

How do you create and activate a virtual environment using Virtualenv?

Answer: To create a virtual environment, use the command `python3 -m venv venv` after creating a directory for your project. To activate this environment, run `source venv/bin/activate`. This isolates your project dependencies, preventing conflicts with other projects.

4.Question

What role does pip play in managing packages for FastAPI applications?

Answer: Pip is a package manager for Python that helps automate the installation, updating, and removal of project dependencies. It enables you to install FastAPI and other necessary libraries easily, simplifying the management of your application's packages.

5.Question

What is Docker, and why is it advantageous for deploying

More Free Books on Bookey



Scan to Download

applications?

Answer:Docker is a tool for containerizing applications, allowing you to package your app and its dependencies into a single container. This ensures consistency across different environments and simplifies deployment, as Docker images can be easily shared and run anywhere.

6.Question

How do you create a simple FastAPI application?

Answer:To create a simple FastAPI application, first install FastAPI and Uvicorn. Then, create an instance of FastAPI in a file (e.g., api.py) and set up routes using decorators.

Finally, run the application using Uvicorn with a command like `uvicorn api:app --port 8000 --reload`.

7.Question

What is the purpose of a .gitignore file, and what common entries should it contain?

Answer:.gitignore is used to specify files and directories that Git should ignore, ensuring sensitive or unnecessary files are not tracked. Common entries include environment files

More Free Books on Bookey



Scan to Download

(.env), the virtual environment folder (env/venv), and IDE metadata folders like .vscode.

8. Question

Explain the process of creating a new branch in Git and its significance.

Answer: To create a new branch in Git, use the command `git checkout -b newbranch`, which creates and switches to the branch. This allows developers to work on features or fixes independently without affecting the main codebase, facilitating collaboration and code management.

9. Question

What are the basic commands you need to run with pip for managing your application dependencies?

Answer: With pip, basic commands include `pip install <package>` to install, `pip uninstall <package>` to remove, `pip freeze > requirements.txt` to list current packages into a file, and `pip install -r requirements.txt` to install packages from that file.

10. Question

After completing Chapter 1, what skills will you have

More Free Books on Bookey



Scan to Download

gained?

Answer: By the end of Chapter 1, you will be able to set up and use Git, manage packages with pip, create isolated development environments with Virtualenv, understand the basics of Docker, and scaffold a simple FastAPI application.

Chapter 2 | Routing in FastAPI| Q&A

1. Question

What is routing in FastAPI?

Answer: Routing in FastAPI refers to the process of handling HTTP requests sent from a client to the server by defining routes with specific handlers to process and respond to those requests.

2. Question

How do you define a route in FastAPI?

Answer: A route is defined by specifying the HTTP method (like GET or POST) and creating a function (route handler) that processes the request. For example, using `@app.get('/example')` defines a GET route that directs to the corresponding handler.

More Free Books on Bookey



Scan to Download

3.Question

What is the purpose of the APIRouter class in FastAPI?

Answer: The APIRouter class allows for the creation of path operations for multiple routes, promoting modularity and organization in routing, especially for larger applications.

4.Question

What are path parameters in FastAPI?

Answer: Path parameters are defined within the URL paths to identify resources. They serve as placeholders in the route definition, enabling operations on specific resources.

5.Question

What is the difference between path parameters and query parameters?

Answer: Path parameters are part of the URL and used to specify which resource to interact with (e.g., `/todo/{todo_id}`), while query parameters appear after a `?` in the URL and are used to provide additional filtering or specifications for the request (e.g., `/todos?status=completed`).

6.Question

More Free Books on Bookey



Scan to Download

Why use validation with Pydantic models in FastAPI?

Answer: Validation with Pydantic models ensures that the data received in requests conforms to a defined structure. This prevents malformed data from being processed and reduces the risk of errors or potential attacks.

7.Question

What is a CRUD application?

Answer: CRUD stands for Create, Read, Update, and Delete - the four basic operations that can be performed on data. A CRUD application allows users to create new records, read existing ones, update them, and delete them.

8.Question

How does FastAPI automatically generate documentation for APIs?

Answer: FastAPI automatically generates OpenAPI documentation for APIs by creating JSON schema definitions for the routes and models defined, which can be accessed via `/docs` for Swagger UI and `/redoc` for ReDoc.

9.Question

What are the roles of the DELETE method in CRUD

More Free Books on Bookey



Scan to Download

operations?

Answer: The DELETE method is used to remove resources from the server, allowing users to delete specific items or all items in a resource.

10. Question

How does one implement error handling in FastAPI?

Answer: Error handling is implemented in FastAPI using status codes and custom exception handlers to manage errors gracefully and provide meaningful feedback to the client.

Chapter 3 | Response Models and Error Handling| Q&A

1. Question

What are response models in FastAPI and why are they important?

Answer: Response models are templates built using Pydantic that define the format of data returned from API routes. They are important because they ensure that the data returned conforms to a specific structure, making it predictable for clients and improving data integrity.

More Free Books on Bookey



Scan to Download

2.Question

What are the components of an API response?

Answer: An API response consists of a header and a body.

The response header contains metadata about the response, such as the status code and Content-Type, while the response body contains the actual data requested by the client.

3.Question

What is the purpose of HTTP status codes in API responses?

Answer: HTTP status codes are issued by a server to indicate the result of a client's request. They categorize responses into groups, providing information about whether a request was successful (2XX), there was a client error (4XX), or there was a server error (5XX).

4.Question

How can one handle errors effectively in a FastAPI application?

Answer: Errors in FastAPI are handled by raising an `HTTPException`, which allows developers to specify the status code and a descriptive error message. This provides

More Free Books on [Bookey](#)



Scan to Download

clear feedback to clients about what went wrong, rather than returning a generic success status.

5. Question

What is the significance of defining response models using Pydantic?

Answer: Defining response models using Pydantic ensures that responses are valid and follow the specified schema. It provides a clear structure for the data returned by the API, which helps in maintaining data consistency and improving the client's experience.

6. Question

Can you explain how the response model changes the output of an API endpoint?

Answer: Using a response model allows developers to specify which fields are returned in the API response. For example, by defining a response model that only includes the 'item' field of to-do items, we can return a cleaner response without exposing unnecessary details like IDs.

7. Question

What steps would you take to implement error handling

More Free Books on Bookey



Scan to Download

in a FastAPI route?

Answer: To implement error handling in a FastAPI route, you can check for conditions that may lead to an error (e.g., resource not found). If such a condition is met, raise an `HTTPException` with an appropriate status code and detail message to inform the client.

8.Question

How did the chapter demonstrate the application of response models and error handling in action?

Answer: The chapter illustrated the use of response models by showing how to define a model for to-do items and return that model from a route. It also demonstrated error handling by updating routes with `HTTPException`, ensuring that the correct status code is returned when a to-do item is not found.

9.Question

What is a practical example of using response models in the code provided?

Answer: In the code provided, the `TodoItems` model is created

More Free Books on Bookey



Scan to Download

to encapsulate a list of TodoItem objects. This is used in the retrieve_todo route to ensure that the response only includes the item names, enhancing API clarity.

10. Question

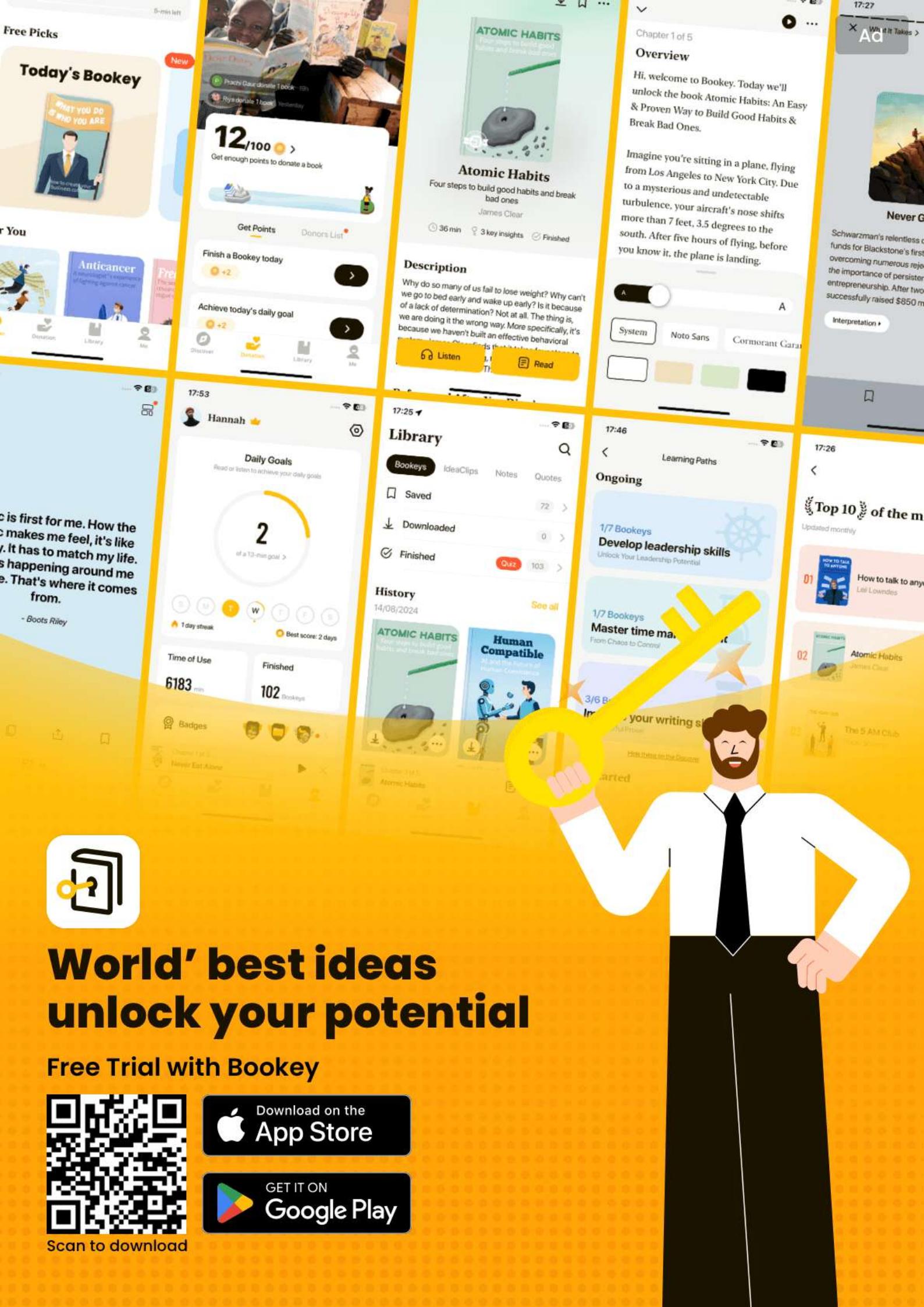
What changes are made to an API endpoint to use the HTTPException for error responses?

Answer: To use HTTPException, you replace the normal return statements with a raise statement that creates an HTTPException instance, specifying the desired status code and an error message.

More Free Books on Bookey



Scan to Download



World's best ideas unlock your potential

Free Trial with Bookey



Scan to download



Download on the
App Store



Chapter 4 | Templating in FastAPI| Q&A

1.Question

What is the purpose of templating in FastAPI?

Answer: Templating in FastAPI serves the purpose of rendering API responses on a web page using templates powered by Jinja. It allows displaying data from the API in various formats, acting as a frontend component for web applications.

2.Question

What role does Jinja play in web development with FastAPI?

Answer: Jinja is a templating engine that facilitates the rendering process of API responses. It uses specific syntax for variable substitution and control structures, which helps to dynamically generate HTML pages based on the data from the API.

3.Question

How does the Jinja templating syntax distinguish between different types of code?

Answer: In Jinja, the syntax uses curly brackets: {{ }} for

More Free Books on Bookey



Scan to Download

variable blocks, { % % } for control structures (like if statements and loops), and {# #} for comments. This clear distinction allows for effective blending of HTML and dynamic content.

4. Question

Can you give an example of how filters work in Jinja?

Answer: Filters in Jinja allow for modifications of variables.

For instance, using the default filter, if a variable is None, it can display a default message: { { todo.item | default('This is a default todo item') } }. If 'todo.item' is None, it will show 'This is a default todo item' instead.

5. Question

What is the significance of template inheritance in Jinja?

Answer: Template inheritance in Jinja allows developers to define a base template that serves as a foundation, which can then be extended or modified by child templates. This follows the DRY (Don't Repeat Yourself) principle, making it efficient for large web applications.

6. Question

How do you install Jinja and set it up with FastAPI?

More Free Books on Bookey



Scan to Download

Answer: To install Jinja, use the command 'pip install jinja2'. Create a directory named 'templates' in your project directory to store all your Jinja files. Configure Jinja in your FastAPI application by using 'from fastapi.templating import Jinja2Templates' and setting up the templates directory.

7. Question

What are some common filters available in Jinja?

Answer: Common Jinja filters include: 'default' to provide default values, 'escape' to render raw HTML output, 'int' and 'float' for type conversion, 'join' to concatenate list elements into a string, and 'length' to get the size of an object.

8. Question

What does a macro in Jinja do?

Answer: A macro in Jinja is similar to a function, designed to return an HTML string. It helps to reuse code without duplication, making it easier to maintain. For example, a macro can streamline the process of creating input fields in HTML forms.

9. Question

What should you include in your templates to render

More Free Books on Bookey



Scan to Download

them correctly with FastAPI?

Answer: In your Jinja templates, include placeholders to render dynamic content (like todo items), leveraging the syntax `{{ variable }}` and control structures like `{% if %}` or `{% for %}` to manage logic and iteration.

10. Question

What is the initial step to begin using Jinja templates in a FastAPI application?

Answer: The first step is to install the Jinja2 package and create a templates directory in your project folder, where you will place your HTML template files that integrate Jinja syntax.

Chapter 5 | Structuring FastAPI Applications| Q&A

1. Question

Why is proper structuring important in FastAPI applications?

Answer: Proper structuring enhances the readability of the application's code, facilitates faster development and debugging, and ultimately

More Free Books on Bookey



Scan to Download

increases overall productivity.

2.Question

What does structuring an application in FastAPI entail?

Answer: Structuring involves organizing application components in a modular format, such as managing routes, models, and database connections in separate folders and files.

3.Question

What are the key components of the event planner application structure described in Chapter 5?

Answer: The key components include a main.py file, and subfolders for database, routes, and models, with specific files for handling events and users.

4.Question

How are event and user models defined in the application?

Answer: The Event model contains fields such as id, title, image, description, tags, and location, while the User model includes email, password, and a list of events, facilitating the organization of related data.

More Free Books on Bookey



Scan to Download

5.Question

What functions do the routes serve in the FastAPI application?

Answer: Routes define the API's endpoints, including functionalities for user registration, sign-in, and event management (create, retrieve, update, delete), enabling users to interact with the application.

6.Question

How does the application handle user registration?

Answer: The application checks for existing users with the same email before registering a new user to ensure uniqueness, returning appropriate messages based on the operation's success or failure.

7.Question

What potential security concern is mentioned regarding password storage in the example?

Answer: The passwords are stored plainly without any encryption, which is considered a bad practice and will be addressed in future chapters focusing on proper database management and security.

More Free Books on Bookey



Scan to Download

8.Question

What steps are involved in testing the routes implemented in the application?

Answer: To test the routes, the application uses cURL commands to simulate API requests, inspecting responses for successful registration, sign-in, event creation, retrieval, and deletion.

9.Question

What is the significance of the interactive documentation generated by FastAPI?

Answer: The interactive documentation, powered by Swagger, allows developers to view and test the API endpoints directly from a web interface, making it easier to understand and utilize the API.

10.Question

What will be covered in the next chapter following Chapter 5's focus on structuring?

Answer: Chapter 6 will introduce connecting the application to SQL and NoSQL databases, expanding the functionality of the event planner application and incorporating persistent

More Free Books on Bookey



Scan to Download

data storage.

Chapter 6 | Connecting to a Database| Q&A

1. Question

What is the primary advantage of connecting a FastAPI application to a proper database compared to an in-app database?

Answer: A proper database allows for permanent data storage, ensuring that data is retained even after application restarts or crashes, unlike an in-app database which loses its data when the application closes.

2. Question

What are SQLAlchemy and Beanie used for in the context of FastAPI applications?

Answer: SQLAlchemy is used for connecting and performing operations on SQL databases, while Beanie is an ODM that facilitates connections and operations with MongoDB.

3. Question

How do you set up a SQL database connection in a FastAPI application?

More Free Books on Bookey



Scan to Download

Answer: You set up a connection by creating a SQLAlchemy engine using the `create_engine()` function from `SQLModel`, providing a database URL, and then calling `SQLModel.metadata.create_all(engine)` to persist the connection.

4. Question

What is a primary key in a database, and why is it important?

Answer: A primary key is a unique identifier for a record in a database table, which is essential for distinguishing each record and ensuring data integrity.

5. Question

Explain how CRUD operations are implemented in FastAPI when using SQL databases.

Answer: CRUD operations (Create, Read, Update, Delete) are implemented via route handlers that utilize a session from `SQLModel` to perform operations such as adding new records, retrieving existing ones, updating records, and deleting entries from the database.

More Free Books on Bookey



Scan to Download

6.Question

In the context of integrating MongoDB with FastAPI, what is a document?

Answer: In MongoDB, data is stored as documents instead of tables; a document is a flexible, JSON-like data structure that defines how data is organized within a collection.

7.Question

How can you ensure that an environment variable for the database URL is used in a FastAPI application?

Answer: By creating a settings class using Pydantic's BaseSettings which reads environment variables from a .env file, ensuring that the DATABASE_URL can be retrieved securely.

8.Question

What role does the Depends class play in FastAPI?

Answer: The Depends class is used for dependency injection, allowing route handlers to specify functions or objects that are required for execution, ensuring that dependencies are met before the handler code runs.

9.Question

More Free Books on Bookey



Scan to Download

What changes must be made to the route handler when transitioning from SQL-based operations to MongoDB-based operations in FastAPI?

Answer: When transitioning, you replace session-based operations with methods provided by the Beanie ODM to manage documents within the MongoDB database, such as `.insert_one()`, `.find()`, and `.delete()`.

10. Question

What should developers expect to learn in the upcoming chapter after this one?

Answer: Developers will learn about securing FastAPI applications, focusing on authentication fundamentals, various methods available in FastAPI, and implementing a JSON Web Token (JWT) system for route protection.

More Free Books on Bookey



Scan to Download

Ad



Scan to Download
Bookey App



Try Bookey App to read 1000+ summary of world best books

Unlock 1000+ Titles, 80+ Topics

New titles added every week

Brand

Leadership & Collaboration

Time Management

Relationship & Communication

Business Strategy

Creativity

Public

Money & Investing

Know Yourself



Positive P

Entrepreneurship

World History

Parent-Child Communication

Self-care

Mind & Sp

Insights of world best books

THINKING,
FAST AND SLOW
How we make decisions



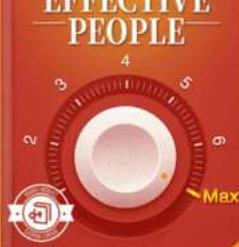
THE 48 LAWS OF POWER
Mastering the art of power, to have the strength to confront complicated situations



ATOMIC HABITS
Four steps to build good habits and break bad ones



THE 7 HABITS OF
HIGHLY
EFFECTIVE
PEOPLE



HOW TO TALK
TO ANYONE
Unlocking the Secrets of
Effective Communication



Free Trial with Bookey



Chapter 7 | Securing FastAPI Applications| Q&A

1.Question

Why is securing an application important?

Answer: Securing an application is crucial to restrict access to its functionalities from unauthorized entities, thereby preventing potential hacks or illegal modifications. It ensures that sensitive data and processes are protected from misuse.

2.Question

What are the key differences between authentication and authorization?

Answer: Authentication is the process of verifying an entity's credentials (such as a username and password), while authorization refers to giving that entity permission to perform specific actions once their identity is confirmed.

3.Question

What authentication methods does FastAPI support?

Answer: FastAPI supports basic HTTP authentication, cookies, and bearer token authentication with a common use case being JWT (JSON Web Token). Each method has

More Free Books on Bookey



Scan to Download

specific use cases and pros and cons.

4.Question

What is dependency injection and how is it used in FastAPI?

Answer: Dependency injection in FastAPI is a design pattern where functions receive instances they need to operate. It allows for cleaner code by injecting dependencies at runtime, making it easier to manage dependencies and reduce code repetition, especially for authentication and authorization.

5.Question

How does JWT enhance the security of a FastAPI application?

Answer: JWT enhances security by ensuring that tokens are signed with a secret key known only to the sender and receiver, preventing tampering. The token's payload can include user information and an expiry time, which improves session handling and access control.

6.Question

What steps were taken to hash user passwords in the FastAPI application?

More Free Books on Bookey



Scan to Download

Answer: To hash user passwords, the `passlib` library was used for bcrypt hashing. A `HashPassword` class was created with methods to create hashes and verify them, ensuring that passwords are stored securely rather than in plain text.

7. Question

What is the role of middleware like CORS in a FastAPI application?

Answer: CORS (Cross-Origin Resource Sharing) middleware serves to manage and restrict which origins are permitted to access resources from the API. It acts as a gatekeeper, allowing for requests from specific domains or from all origins, depending on the configuration.

8. Question

How were routes updated to restrict access to authenticated users in the application?

Answer: Routes were updated by injecting the authentication dependency into POST, PUT, and DELETE operations for creating, updating, and deleting events. This ensured that only users with valid tokens could access these routes,

More Free Books on Bookey



Scan to Download

protecting sensitive operations.

9.Question

What did the summary of the chapter emphasize about the key concepts learned?

Answer: The summary emphasized the importance of securing FastAPI applications using OAuth and JWT, understanding dependency injection, protecting routes from unauthorized access, and configuring CORS. These skills serve as foundational knowledge for further development and testing of FastAPI applications.

Chapter 8 | Testing FastAPI Applications| Q&A

1.Question

Why is testing considered an integral part of the application development cycle?

Answer: Testing is essential because it ensures that the application functions as expected and detects anomalies before the application is deployed to production, which helps maintain reliability and user satisfaction.

More Free Books on Bookey



Scan to Download

2.Question

What is unit testing, and why is it important?

Answer: Unit testing is a procedure where individual components of an application are tested to verify their functionality. It is important because it allows developers to ensure that specific parts of the code operate correctly, reducing the likelihood of defects in the final product.

3.Question

How does pytest differ from Python's built-in unittest module?

Answer: Pytest offers a shorter syntax and more flexible features than unittest, making it a preferred choice for testing Python applications due to its ease of use and powerful functionality.

4.Question

What role do fixtures play in pytest?

Answer: Fixtures in pytest are reusable functions that return data needed by test functions. They help eliminate redundancy by allowing the setup of data or state that multiple tests can share.

More Free Books on Bookey



Scan to Download

5.Question

What steps need to be taken to set up a test environment for FastAPI applications?

Answer: To set up a test environment for FastAPI applications, ensure a running MongoDB server, install required Python libraries such as `httpx` and `pytest-asyncio`, and create a configuration file for `pytest` to enable asynchronous testing.

6.Question

How are tests written for specific endpoints in a FastAPI application?

Answer: Tests for specific endpoints are written by creating test functions that send requests to those endpoints and assert the responses against expected outcomes, utilizing fixtures to manage state and setup.

7.Question

Why is it beneficial to generate a coverage report for tests?

Answer: Generating a coverage report provides insights into which portions of the code were executed during testing,

More Free Books on Bookey



Scan to Download

helping identify untested code sections that may require additional tests to improve application robustness.

8. Question

What does the assert statement do in the context of pytest?

Answer: The assert statement in pytest is used to verify that a specific condition holds true. If the condition is false, pytest will raise an `AssertionError` and signal that the test has failed.

9. Question

What are the purposes of the `@pytest.mark.asyncio` decorator in tests?

Answer: The `@pytest.mark.asyncio` decorator indicates that the test function is asynchronous, allowing pytest to run the test function in an event loop that is necessary for testing asynchronous code.

10. Question

What should you do if a test fails, according to the content?

Answer: If a test fails, you should check the failing assertion statement to understand the discrepancy and correct either the

More Free Books on Bookey



Scan to Download

test or the application code accordingly, ensuring the expected behavior is accurately defined.

Chapter 9 | Deploying FastAPI Applications| Q&A

1.Question

What is the first step in preparing a FastAPI application for deployment?

Answer: The first step is to ensure that the application dependencies are up to date in the `requirements.txt` file.

2.Question

Why is it important to maintain the `requirements.txt` file?

Answer: Maintaining the `requirements.txt` file allows for better management of application dependencies and ensures that only the necessary packages are included, making installation and deployment smoother.

3.Question

What command is used to retrieve a list of installed packages in Python?

Answer: The `pip freeze` command is used to retrieve a list of

More Free Books on Bookey



Scan to Download

all packages installed in the development environment.

4.Question

How do we configure environment variables for the FastAPI application?

Answer: Environment variables can be injected during deployment and should be kept out of version control systems such as GitHub for security.

5.Question

What is Docker, and why is it useful for deploying FastAPI applications?

Answer: Docker is a containerization technology that allows applications to run in self-contained environments, making it easier to manage dependencies and ensure that applications work across different environments.

6.Question

What is the purpose of a Dockerfile?

Answer: A Dockerfile contains a set of instructions for building a Docker image, which can then be deployed to run applications.

7.Question

More Free Books on Bookey



Scan to Download

What command is used to build a Docker image?

Answer: The command `docker build -t <image-name> .` is used to build a Docker image from the instructions defined in the current directory.

8.Question

What is a docker-compose file, and what does it do?

Answer: A docker-compose file (usually named `docker-compose.yml`) defines and manages multi-container Docker applications, allowing services to be easily started, stopped, and configured.

9.Question

How do you verify that the FastAPI application is running successfully after deployment?

Answer: You can send a GET request to the application's endpoint (e.g., `http://localhost:8080/event/`) and check the response.

10.Question

What command is used to stop the services in a Docker deployment?

Answer: The command `docker-compose down` is used to

More Free Books on Bookey



Scan to Download

stop the services.

11. Question

What are the benefits of using Docker images for deploying applications?

Answer: Docker images provide consistency across different environments, reduce runtime issues related to dependencies, and enable easy scaling and deployment to various platforms.

12. Question

What alternative is suggested for hosting MongoDB when deploying to serverless platforms?

Answer: For serverless platforms that do not support deploying Docker containers for databases, it is suggested to host the MongoDB database on MongoDB Atlas.

13. Question

What is a .dockerignore file, and why is it used?

Answer: A .dockerignore file specifies files and directories to exclude from the Docker image build context, reducing the size of the Docker image and avoiding unnecessary files.

14. Question

What is the `CMD` command in a Dockerfile?

More Free Books on Bookey



Scan to Download

Answer: The `CMD` command specifies the default command to run when a container is started from the Docker image.

More Free Books on Bookey



Scan to Download



Scan to Download



Why Bookey is must have App for Book Lovers

30min Content



The deeper and clearer interpretation we provide, the better grasp of each title you have.



Text and Audio format

Absorb knowledge even in fragmented time.



Quiz

Check whether you have mastered what you just learned.



And more

Multiple Voices & fonts, Mind Map, Quotes, IdeaClips...

Free Trial with Bookey



Building Python Web Apis With Fastapi

Quiz and Test

Check the Correct Answer on Bookey Website

Chapter 1 | Getting Started with FastAPI| Quiz and Test

1. FastAPI is a modern Python web framework that is faster and easier to learn compared to Flask and Django.
2. Virtualenv is used to create isolated environments to allow conflicts from globally installed packages.
3. To activate a virtual environment, you should use the command `pip install fastapi`.

Chapter 2 | Routing in FastAPI| Quiz and Test

1. FastAPI requires a complex setup for routing applications.
2. The APIRouter class helps in organizing multiple routes to enhance code readability.
3. Path parameters are used in the body of a request to identify resources.

More Free Books on Bookey



Scan to Download

Chapter 3 | Response Models and Error Handling| Quiz and Test

1. Response models in FastAPI are built using Pydantic and serve as templates for returning structured data.
2. Error handling in FastAPI is optional and can be ignored during API development.
3. HTTP status codes categorize responses into various groups such as 1XX for request received and 4XX for client errors.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1000+ Book Summaries with Quizzes

Free Trial Available!

Scan to Download



Download on the
App Store

GET IT ON
Google Play

Atomic Habits
Four steps to build good habits and break bad ones
James Clear
🕒 36 min ⚡ 3 key insights ✅ Finished

Description

Why do so many of us fail to lose weight? Why can't we go to bed early and wake up early? Is it because of a lack of determination? Not at all. The thing is, we are doing it the wrong way. More specifically, it's because we haven't built an effective behavioral pattern. James Clear finds that it takes four steps to...

6 Listen 11 Read 1 Th...

10:16 X 1 of 5

Habit building requires four steps: cue, craving, response, and reward are the pillars of every habit.

False **True**

10:16 X 5 of 5

The Two-Minute Rule is a quick way to end procrastination, but it only works for two minutes and does little to build long-term habits.

False

Correct Answer

Once you've learned to care for the seed of every habit, the first two minutes are just the initiation of formal matters. Over time, you'll forget the two-minute time limit and get better at building the habit.

Continue

Chapter 4 | Templating in FastAPI| Quiz and Test

- 1.Jinja is a templating engine that allows rendering API responses in FastAPI.
- 2.Control structures in Ninja are indicated by the `{{ }}` syntax.
- 3.Macros in Ninja are used to provide a fallback value if a variable is None.

Chapter 5 | Structuring FastAPI Applications| Quiz and Test

- 1.In FastAPI applications, structuring routes and models properly can enhance readability and increase productivity.
- 2.The Event Planner application allows users to only create events without any ability to update or delete them.
- 3.Models in FastAPI are used to outline data storage and representation including fields like id, title, and description.

Chapter 6 | Connecting to a Database| Quiz and Test

- 1.In Chapter 6, it is stated that the purpose of integrating a database into a FastAPI application

More Free Books on Bookey



Scan to Download

is to enable persistent data storage, where data is retained even after app restarts or crashes.

2. The chapter explains how to operate with SQL databases using a library called MongoDB.
3. Setting up CRUD operations for both SQL and NoSQL is covered in Chapter 6 of the book.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1000+ Book Summaries with Quizzes

Free Trial Available!

Scan to Download



Download on the
App Store

GET IT ON
Google Play

Atomic Habits
Four steps to build good habits and break bad ones
James Clear
🕒 36 min ⚡ 3 key insights ✅ Finished

Description

Why do so many of us fail to lose weight? Why can't we go to bed early and wake up early? Is it because of a lack of determination? Not at all. The thing is, we are doing it the wrong way. More specifically, it's because we haven't built an effective behavioral pattern. James Clear finds that it takes four steps to...

6 Listen 11 Read 1 Th...

10:16

1 of 5

Habit building requires four steps: cue, craving, response, and reward are the pillars of every habit.

False True

10:16

5 of 5

The Two-Minute Rule is a quick way to end procrastination, but it only works for two minutes and does little to build long-term habits.

False

Correct Answer

Once you've learned to care for the seed of every habit, the first two minutes are just the initiation of formal matters. Over time, you'll forget the two-minute time limit and get better at building the habit.

Continue

Chapter 7 | Securing FastAPI Applications| Quiz and Test

1. Authenticating a user means verifying their credentials.
2. FastAPI does not support bearer token authentication as a method of securing an application.
3. Dependency Injection is a method used in FastAPI to separate concerns and manage dependencies during request handling.

Chapter 8 | Testing FastAPI Applications| Quiz and Test

1. Unit testing is unnecessary for FastAPI applications.
2. Fixtures in pytest can return necessary data for tests and simplify test setup.
3. HTTPX is used for testing synchronous APIs in FastAPI applications.

Chapter 9 | Deploying FastAPI Applications| Quiz and Test

1. Before deploying a FastAPI application, the

More Free Books on Bookey



Scan to Download

requirements.txt file must be updated to reflect the current application's dependencies.

2. It's acceptable to include environment variables in version control systems when deploying FastAPI applications.
3. Docker allows applications to run consistently across different environments by enabling containerization.

More Free Books on Bookey



Scan to Download



Download Bookey App to enjoy

1000+ Book Summaries with Quizzes

Free Trial Available!

Scan to Download



Download on the
App Store

GET IT ON
Google Play

Atomic Habits
Four steps to build good habits and break bad ones
James Clear
🕒 36 min ⚡ 3 key insights ✅ Finished

Description

Why do so many of us fail to lose weight? Why can't we go to bed early and wake up early? Is it because of a lack of determination? Not at all. The thing is, we are doing it the wrong way. More specifically, it's because we haven't built an effective behavioral pattern. James Clear finds that it takes four steps to...

6 Listen 11 Read 1 Th...

10:16

1 of 5

Habit building requires four steps: cue, craving, response, and reward are the pillars of every habit.

False True

10:16

5 of 5

The Two-Minute Rule is a quick way to end procrastination, but it only works for two minutes and does little to build long-term habits.

False

Correct Answer

Once you've learned to care for the seed of every habit, the first two minutes are just the initiation of formal matters. Over time, you'll forget the two-minute time limit and get better at building the habit.

Continue