Ankita Mate

Que 1. What is FastAPI and what are its main features?

Ans: FastAPI is a Python web framework designed for building APIs.

* Features of Fast API
* Automatic data validation
* Automatic documentation
* Asynchronous support
* Security features: Supports OAuth2 and JWT
* Dependency injection

Que 2. How do you install FastAPI?

Ans: 1. install by command pip install fastapi

2 then install uvicorn by command pip insatlll uvicorn . uvcorn is ASGI

server necessary to run fast API application

Que 3. What makes FastAPI different from other Python web frameworks?

Ans:

* [FastAPI surpasses Flask in terms of performance](https://fastapi.tiangolo.com/#performance/), and it is one of the fastest Python web frameworks. Only [Starlette](https://www.starlette.io/) and [Uvicorn](https://www.uvicorn.org/" \t "_blank) are faster. Because of ASGI, FastAPI supports concurrency and asynchronous code by declaring the endpoints.
* FastAPI simplifies concurrency by eliminating the need for an event loop or async/await management
* FastAPI supports a dependency injection solution that is simple and easy to use.
* Built-in data validation enables developers to omit proof and write more compact code.

Que 4. What is asynchronous programming, and how does FastAPI utilize it?

Ans: Asynchronous programming allows a program to run multiple tasks

concurrently, without waiting for each task to finish before starting the

next.

FastAPI uses the async and await syntax to define asynchronous

code

Que 5. How do you create a FastAPI application?

Ans:

* Create virtual environment
* Activate virtual environment
* Install FastAPI by using command pip install fastapi
* Install uvicorn to run fast API application by the command pip install uvicorn
* Create a python file called any name
* Import fastApi
* Create a FastAPI "instance
* create a path operation. Path. Operation. Define a path operation decorator.
* define the path operation function
* return the content.

Here is an Example:

from fastapi import FastAPI

app = FastAPI()

@app.get("/")

async def root():

return {"message": "Hello World"}

run the server

Que 6. Explain the concept of path parameters and query parameters in FastAPI.

Ans**: Path parameters** are components of a URL path that can vary

**query parameters** are optional parameters in a URL, which mostly

appear after the question mark**(?)**, and are separated by the and **(&)**

symbol,

Que 7. What are request and response models in FastAPI? How are they defined?

Ans: In FastAPI, request and response models define the structure of

Que 8. How do you define route operations in FastAPI?

Ans: In FastAPI, you can define route operations by using decorators to associate functions with specific URLs and HTTP methods. For example, you can use the @app.get() decorator to define a route that returns items:

Que 9. What is dependency injection in FastAPI? How is it used?

Ans: Dependency injection is a design pattern that allows code to declare its dependencies, and the system, in this case FastAPI, provides them. This decoupling of code from its dependencies improves reusability, maintainability, and testability

data that's sent and received by an API. They're typically Pydantic

models, which offer data validation, serialization, and

documentation generation.

Request model: Defines the data structure of the request body sent by

the client

Response model: Defines the data structure of the response body sent

back to the client