Top 40 Flask Interview Questions and Answers

Last Updated : 15 Jun, 2024

Flask is a popular Python web framework used to build web applications. If you're preparing for a Flask development position, it's important to be ready for the types of questions that you might encounter in an interview. In this article, we'll go through some of the top 40 Flask interview questions and provide answers to help you prepare.

Flask Interview Questions And Answers

Q 1. What is Flask?

Answer: Flask is a microweb framework that provides an API to build up web applications. Flask's framework is also easier to learn because of its diversified working style. Flask is based on the WSGI (Web Server Gateway Interface) toolkit and the Jinja2 template engine. It is very flexible to implement a simple web application. Also, Flask provides visual debugging, which gives more control over the component.

Q 2. What are the features of Flask Python?

Answer:

- Built-in web server and debugger
- Compatibility with most of the latest technologies.
- High scalability and flexibility for simple web applications.
- Integrated support for unit testing
- Securing cookies in client-side sessions
- Dispatching RESTful request
- Google App Engine compatibility
- Unicode support
- Web Server Gateway Interface(WSGI) compliance

Q 3. What is the difference between Flask and Django?

Answer:

Flask	Django
Flask is a WSGI framework	Django is a Full-stack web framework
It allows multiple types of databases.	It doesn't support multiple types of databases.
Use SQL Alchemy	Build-in ORM
Diversified Working Style	Monolithic Working Style
Arbitrary structure	Conventional Project Structure
It supports API	It does not have any support for API
It does not support Dynamic HTML pages	Django accepts Dynamic Pages.

It has support for Visual debug	No support for Visual Debug
It doesn't offer a built-in bootstrapping tool.	Django-admin enables us to start building web applications without any external input,
URL dispatcher is a RESTful request.	URL dispatcher is Robust Documentation.

Q 4. What is the default host port and port of Flask?

Answer: The default local host of the flask is 127.0.0.1, and the default port is 5000.

Q 5. Which databases are Flask compatible with?

Answer: As a backend database, Flask supports SQLite and MySQL. DbAdapters are used to support various databases. It comes with an SQLDbAdapter that allows you to connect to a variety of SQL databases using Flask-SQLAlchemy, including MySQL, Oracle, PostgreSQL, SQLite, Sybase, Firebird, and others. It includes a MongoDbAdapter that allows you to connect to MongoDB databases using Flask-MongoEngine.

Q 6. why do we use Flask(__name__) in Flask?

Answer: The __name__ parameter is a Python built-in variable that is set to the name of the current module. When we pass __name__ as an argument to the Flask class constructor, it helps Flask to determine where to locate resources such as templates and static files.

Q 7. What is routing in Flask?

Answer: App Routing means mapping the URLs to a specific function that will handle the logic for that URL. Modern web frameworks use more meaningful URLs to help users remember the URLs and make navigation simpler.

So if our site's domain was www.example.org and we want to add routing to "www.example.org/hello", we would use "/hello".

Q 8. What is Template Inheritance in Flask?

Answer: Template Inheritance is a powerful feature of Flask's Jinja templating has a great feature called template inheritance. Jinja is a Python programming language web template engine. We've noticed that a website's web pages all have the same footer, navigation bar, and other elements. Instead of creating the identical footer and navigation bar on each webpage separately, we utilize template inheritance, which allows us to generate the part that is common to all webpages (e.g. footer, navigation bar) only once and eliminates the need to write the HTML, head, and title tag many times.

Q 9. What does url_for do in Flask?

Answer: The url_for() method is used to generate a URL to a specific function dynamically. After the first argument, which is the name of the selected function, we can send any number of keyword arguments matching the variable portion of the URL. This function is useful since it allows us to create URLs dynamically instead of hard-coding them into the templates.

```
<a href="{{ url_for('get_post_id', post_id=post.id}}">{{post.title}}<a>
```

View function for handling variables in routes.

```
@app.route("/blog/post/<string:post_id>")
def get_post_id(post_id):
    return post_id
```

Q 10. How do you handle cookies in a Flask?

Answer: The set_cookie() method on the response object in Flask is used to set cookies. The view function's make response() method can be used to construct the response object. On the client's PC, cookies are stored as text files. Cookies are used to track a user's online actions and to provide recommendations based on the user’s preferences in order to improve the user’s online experience. Cookies are stored on the client’s machine by the server and are associated with the client’s request to that server in all subsequent transactions until the cookie’s lifetime expires or the cookie is erased by the server’s specific web page.

Q 11. How does file uploading work in Flask?

Answer: The process of sending binary or regular files to a server is known as file uploading. Flask makes it simple for us to upload files. All we need is an HTML form with multipart/form-data encryption enabled. The request.files[] Object is used by the server-side flask script to get the file from the request object. The file is saved to the chosen location on the server when it has been successfully uploaded. You can get the name of the target file by doing the following.

```
request.files['file'] = name.filename
```

Q 12. What is Flask-WTF, and what are its characteristics?

Answer: WTF, also known as WT Forms in Flask, is a type of interactive user interface. The WTF is a flask built-in module that lets you build forms in a different way in flask web apps. Flask-WTF is designed to be simple to connect with WTForms, and it works well with Flask-WTF. Flask WTF includes the following features:

- Integration with web forms is available.
- It comes with a CSRF token, it’s an extremely secure form.
- CSRF protection on a global scale
- Comes with the ability to integrate internationalization.
- There’s also a Supporting Captcha
- This module has a file uploader that works with Flask Uploads.

Q 13. How long can an identifier be in Flask Python?

Answer: In Flask Python, An identifier can be as long as you want, as python is case-sensitive so it will treat upper case and lower case letters differently. Also, there are some words that are reserved keywords that can't be used by users. Some of them are listed below:

> ***def, false, import, not, true, as, del, finally, in, or, try, assert, elseif, for, is, pass, while, break, else, from, lambda, print, with, class, except, global, none, raise, yield, continue, exec, if, nonlocal, return***

There are also some standards that users must follow while naming an identifier. It should start with a character, underscore, or a letter from A to Z or a-z, and the remaining characters in the identifier's name can be any of the following: A-Z or a-z, 0-9, or.

Q 14. What HTTP methods does Python Flask provide?

Answer: To handle HTTP requests, Flask uses a number of decorators. The HTTP protocol is the backbone of internet data communication. This HTTP protocol defines a number of techniques for obtaining information from a particular URL. The different HTTP methods are:

Request	Purpose
GET	The most widely used approach. The server responds with data after receiving a GET message.
POST	To submit HTML form data to the server, use this method. The server does not save the data supplied via the POST method.
PUT	Upload content to replace all current representations of the target resource.
DELETE	Deletes all current representations of the URL's target resource.

Request	Purpose
HEAD	Retrieves the headers for a resource, without retrieving the resource itself.

Q 15. In Flask, what do you mean by template engines?

Answer: Template engines are used when we want to build web applications that are split into different components. It is used for server-side applications that are not created as APIs and run on a single server. Templates also make it possible to render the server-side data that must be provided to the application quickly, such as the body, navigation, footer, dashboard, and so on.

Ejs, Jade, Pug, Mustache, HandlebarsJS, Jinja2, and Blade are some popular template engines.

Q 16. What is the use of jsonify() in Flask?

Answer: Jsonify is one of the flask.json module's functions. It converts data to JSON and encapsulates it in a response object with the mime-type application/JSON. It loads directly from the flask module instead of the flask itself. To put it another way, jsonify() is a Flask helper method for correctly returning JSON data. The application/JSON mime-type is set by jsonify(), whereas json.dumps() just deliver a JSON data string. This could have unanticipated ramifications.

The jsonify() function is useful in Flask apps because it automatically sets the correct response headers and content type for JSON responses, and allows you to easily return JSON-formatted data from your route handlers. This makes it easier and more convenient to create APIs that return JSON data.

Q 17. Explain How You Can Access Sessions In Flask?

Answer: The duration between when a client signs in and logs off of a server is referred to as a session. Flask session is a flask utility that gives server-side support for sessions in the flask application developed. It is a plugin that gives your application server-side session capability. The data that must be saved in the session is saved in a temporary directory on the server. When we need to save a significant amount of data between queries in Flask, we can use session objects.

```
from flask import Flask, render_template, redirect, request, session
# The Session instance is not used for direct access, you should always
use flask.session
from flask_session import Session
app = Flask(__name__)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        session["name"] = request.form.get("name")
        return redirect("/")
    return render_template("login.html")
@app.route("/logout")
def logout():
    session["name"] = None
    return redirect("/")
```

Q 18. Explain how one can one-request database connections in Flask?

Answer: Creating and closing database connections all the time is very inefficient, Because database connections encapsulate a transaction, you must ensure that the connection is only used by one request at a time. The Flask framework allows its users to request databases in three ways. They are:

- **before_request():** No parameters are given when these connections are invoked before making a request.
- **after_request():** After initiating a request, these connections are called, and a response is sent to the client.
- **teardown_request():** This decorator is called when an exception is raised or everything went well (the error parameter will be None).

Q 19. What is the g object? What distinguishes it from the session object?

Answer: g is a global namespace that can hold any data for a single app context. For example, a prior request handler might set g.user, which the route and other functions can access. In the flask, on the other hand, the session data is tracked using a session object, which is a dictionary object that has a key-value pair of the session variables and their associated values. We can save data for a particular browser using the session. The session data is carried over when a user of our Flask app performs more queries in the same browser.

Q 20. Mention how one can enable debugging in Flask Python?

Answer: When Debug is turned on, any changes to the application code are updated immediately in the development stage, eliminating the need to restart the server.

- By setting the flag on the applications object
- Bypassing the flag as a parameter to run. If the user enables debug support, the server will reload it when the code will change and the user doesn't have to restart after each change made in the code.

```
#Method 1
app.debug = True

#Method 2
app.run('host' = localhost, debug = True)
```

Q 21. What do you mean by the Thread-Local object in Flask Python?

Answer: A thread-local object is one that is connected to the current thread id and saved in a specialized structure. Internally, Flask Python uses thread-local objects so that the user does not have to transmit objects from one function to the next within a request to stay thread safe.

Q 22. How is memory managed in Flask Python?

Answer: In a flask, Memory allocation is managed by the Flask Python memory management.   Also, It has an inbuilt garbage collector which recycles all unused memory to save up heap space. The Python interpreter's responsibility is to keep track of everything. Users can, however, use the core API to access some of the tools.

Q 23. What type of Applications can we create with Flask?

Answer: We may develop nearly any form of web application by utilizing Flask. It is so versatile and adaptable that it may be rapidly merged with other technologies. Flask, for example, can be used in concert with NodeJS serverless, AWS lambda, and other third-party services to create cutting-edge systems. We can also build Single Page Apps, RESTful API-based Apps, SAS Apps, Small to Medium Websites, Static Websites, Machine Learning Applications, Microservices, and Serverless Apps.

Q 24. How to create a RESTful application in Flask?

Answer: Flask Restful is a Flask plugin that allows you to create REST APIs in Python using Flask as the backend. To create a REST API, we have to do the following steps:

- Import the modules and start up the program.
- Creating the REST API endpoints
- Define the request methods
- Implement the endpoint handlers
- Serialize Data
- Error Handling
- Test the endpoints using various tools like Postman

Q 25. What Is Flask Sijax?

Answer: Sijax is a Python/jQuery library that makes AJAX easy to use in web applications to your Flask applications.Flask Sijax also provides an easy way to send JSON data between the server and the client.

To install we can use the following command

```
pip install flask-sijax
```

Q 26. Why is Flask called a Microframework?

Answer: Flask is termed “micro” because its main feature set is relatively limited: routing, request processing, and blueprint modules are all there is to it. Many capabilities, such as ORM, caching, and authentication, were available as optional extensions, but competing frameworks (such as Django) included them by default. The “small core + extensions” design makes it a “micro-” framework that is much easier to get started with and scale up.

Q 27. How to get a visitor IP address in Flask?

Answer: To get the visitor IP address in Flask we use method request.remote_addr Below is the implementation of it:

```
from flask import Flask, request

app = Flask(__name__)

@app.route('/')
def get_visitor_ip():
    visitor_ip = request.remote_addr
    return f"Visitor's IP address is: {visitor_ip}"

if __name__ == '__main__':
    app.run(debug=True)
```

Q 28. Which extension is used to connect to a database in Flask?

Answer: Extension enhances database management and interaction during development by eliminating the requirement to write raw SQL queries. PostgreSQL, SQLite, and MySQL are just a few of the RDBMSs Flask supports. The Flask-SQLAlchemy plugin is required to connect to databases.

Q 29. What is logging in to Flask?

Answer: Flask logging gives a lot of capability and flexibility to Flask application developers. It also allows developers to construct a sophisticated, event-logging system for Flask apps and includes all of the essential functions and classes. The same standardized Python logging framework is used in Flask. During logging, the Python modules can communicate and contribute.

Q 30. Explain Application Context and Request Context in Flask?

Answer:

Application Context :

The Application Context is the context in which the Flask application runs. It is created when the application starts and is destroyed when the application when us down. The application context stores the configuration and another global state of the application.

Request Context :

The Request Context is the context in which a request is processed. It is created when a request comes in and is destroyed when the request is completed. The Request Context stores information about the current request, such as the request method, URL, headers, and form data.

Q 31: What is Flask-SocketIO?

Answer: Flask-SocketIO is a Flask extension that provides real-time communication between clients and servers using WebSockets.

Q 32. What is Flask-Bcrypt?

Answer: Flask-Bcrypt is a Flask extension that provides password hashing and verification functionality for Flask applications.

Q33. What is Flask-JWT?

Answer: Flask-JWT is a Flask extension that provides JSON Web Token (JWT) authentication and authorization functionality for Flask applications.

Q 34. What is Flask-Assets?

Answer: Flask-Assets is a Flask extension that provides tools for managing and compiling static assets like CSS and JavaScript files.

Q 35. What is Flask-Migrate?

Answer: Flask-Migrate is a Flask extension that provides database migration functionality for Flask applications.

Q 36. What is Flask-Admin?

Answer: Flask-Admin is a Flask extension that provides a simple interface for building administrative interfaces for Flask applications. It allows you to quickly and easily create CRUD (Create, Read, Update, Delete) interfaces for your application's models and data.

Q 37. What is Flask-SQLAlchemy?

Answer: Flask-SQLAlchemy is a Flask extension that provides an easy-to-use interface for working with SQL databases in Flask applications.

Q 38. How do you handle errors in Flask?

Answer: You can handle errors in Flask by using Flask's error-handling functionality. This allows you to define custom error pages and handlers for different types of errors.

Q 39. What is a Flask blueprint?

Answer: A Flask blueprint is a way to organize your Flask application into smaller, modular components. Blueprints can define routes, templates, and static files, and can be registered with an application to create a larger, more complex application.

Overall, a successful Flask interview will test your knowledge of Flask and its associated technologies, as well as your ability to solve problems and work collaboratively. By reviewing common Flask interview questions and answers, you'll be better prepared to demonstrate your expertise in this popular Python web framework.

Q 40: What is Flask-RESTful?

Answer: Flask-RESTful is a Flask extension that provides tools for building RESTful APIs in Flask applications.

Previous Article **Next Article**

Walmart Labs Interview Experience | Set 3 (On-Campus)

Similar Reads

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is know...

8 min read

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses...

4 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to...

4 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would...

12 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used f...

11 min read

Active Directory Interview Questions - Top 50+ Questions and Answers for…

Active Directory (AD) is a crucial component of modern enterprise IT infrastructure, providing centralized authentication, authorization, and directory...

15+ min read

Teacher Interview Questions - Top 70 Questions and Answers for 2024

Teaching is a noble profession that requires a unique blend of knowledge, skills, and passion. As educators, teachers play a crucial role in shaping the minds of...

15+ min read

Top 50 Blockchain Interview Questions and Answers

Blockchain is one of the most trending technologies out there in the tech world. It is basically concerned with a distributed database that maintains the records of...

15+ min read

Top 50 Solidity Interview Questions and Answers

Solidity is an object-oriented programming language used to implement smart contracts on blockchain platforms like Ethereum, which generates transaction...

15+ min read

Top Interview Questions and Answers on Bubble Sort

Bubble Sort is a basic sorting technique that compares adjacent elements and swaps them if they are in the wrong order. Knowing about Bubble Sort is...

5 min read

Article Tags : Interview Questions Python interview-preparation interview-questions +1 More

Practice Tags : python

DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects