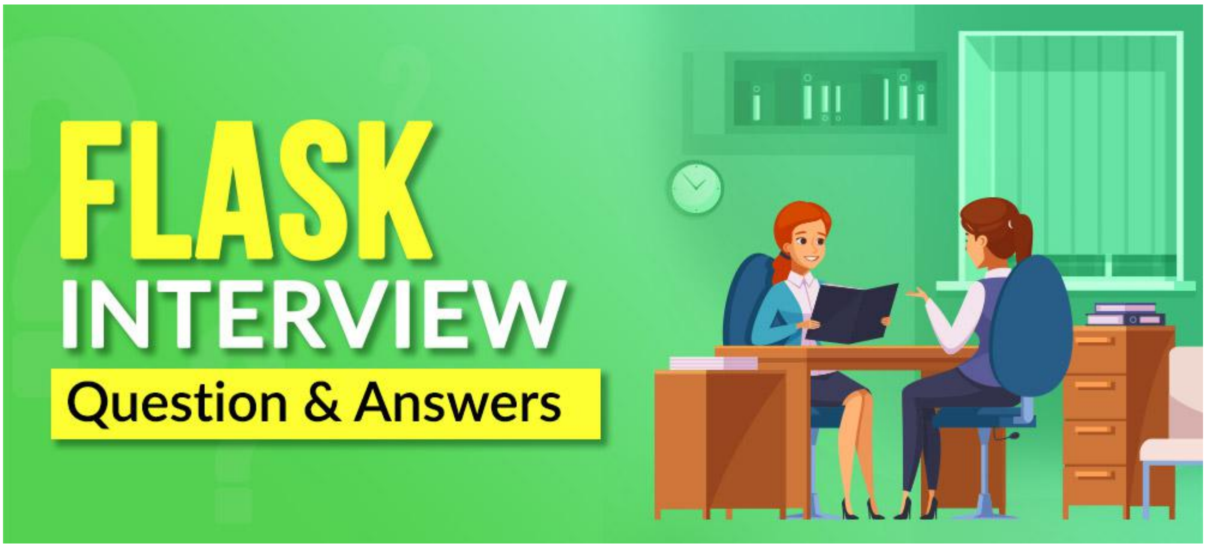
Top 40 Flask Interview Questions and Answers

Read Discuss Courses Practice

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:

- 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.

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
```

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

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)
```