69 Flask interview questions to ask developers (with answers)

Questions

1. undefined
2. undefined
3. undefined
4. undefined
5. undefined
6. undefined
7. undefined
8. How do you create a simple 'Hello, World!' application in Flask?
9. Explain the concept of templates in Flask and how to use them.
10. What is the purpose of the `render_template()` function in Flask?
11. How do you handle form submissions in Flask?
12. Explain the difference between GET and POST methods in Flask routes.
13. What are Flask extensions and can you name a few commonly used ones?
14. How do you set up a database connection in a Flask application?
15. Explain the concept of blueprints in Flask and when you might use them.
16. How do you implement user authentication in a Flask application?
17. What is the purpose of the `request` object in Flask?
18. How do you handle file uploads in Flask?
19. Explain the concept of sessions in Flask and how to use them.
20. What is Flask-WTF and how does it simplify form handling?
21. How do you implement RESTful APIs using Flask?
22. Explain the purpose of the `app.config` object in Flask.
23. How do you handle CORS (Cross-Origin Resource Sharing) in Flask?
24. What is Jinja2 and how is it used in Flask?
25. How do you deploy a Flask application to a production environment?
26. Explain the concept of middleware in Flask and provide an example.
27. How do you implement custom error pages in Flask?
28. How do you ensure scalability in a Flask application?
29. What are some ways to improve the performance of a Flask application?
30. How do you handle session management in Flask?
31. What is the role of middleware in a Flask application?
32. How do you handle asynchronous tasks in Flask?
33. How do you secure a Flask application?
34. What are the benefits of using Flask for microservices architecture?
35. How do you handle logging in a Flask application?
36. How do you manage dependencies in a Flask project?
37. What are some best practices for structuring a Flask project?
38. Can you explain how Flask's context globals, such as `g` and `current_app`, work and when you might use them?
39. How would you implement background tasks in Flask, and what are some common use cases for this?
40. Describe how you would handle real-time updates in a Flask application. What technologies or libraries would you use?
41. Explain the concept and use cases of Flask signal in an application. How would you implement them?
42. How do you manage testing in a Flask application? Can you describe your approach to unit and integration testing?
43. What strategies would you use to optimize the performance of a Flask application?
44. Can you discuss how you would handle database migrations in Flask? What libraries or tools would you utilize?
45. How do you implement role-based access control (RBAC) in a Flask application?
46. Explain the purpose and use of Flask's `before_request` and `after_request` decorators.
47. Describe how you would secure a Flask REST API. What are some potential threats and how would you mitigate them?
48. How would you design a scalable architecture for a large Flask application?
49. Can you explain the advantages and limitations of using Flask in microservices architecture?
50. How do you handle internationalization (i18n) and localization (l10n) in a Flask application?
51. What are some best practices for error handling and logging in a Flask application?
52. Can you discuss how to use Flask with a front-end framework like React or Angular?
53. How do you define a route in Flask and why is it important?
54. What is the role of URL parameters in Flask routing?
55. How would you handle a situation where you have multiple routes pointing to the same function?
56. Can you explain the concept of 'route protection' in Flask?
57. What are some best practices for organizing routes in a large Flask application?
58. How do you handle route conflicts in Flask?
59. What is the significance of the `methods` parameter in Flask routing?
60. How do you ensure that your Flask routes are both SEO-friendly and user-friendly?
61. undefined
62. undefined
63. undefined
64. undefined
65. undefined
66. undefined
67. undefined
68. undefined