Backend & Database Interview Questions

Basics

1. What is Node.js?
2. How does Node.js differ from traditional web servers like Apache?
3. Explain the characteristics of Node.js that make it suitable for building scalable applications.
4. What is npm? How do you use it in Node.js projects?
5. What are the advantages of using Node.js?
6. What is event-driven programming in Node.js?

Modules and Packages

7. How do you include external libraries in Node.js?
8. What is the difference between `require` and `import` in Node.js?
9. How do you create and publish your own npm package?
10. What are the built-in modules in Node.js?

Asynchronous Programming

11. Explain the concept of non-blocking I/O in Node.js.
12. How does Node.js handle asynchronous operations?
13. What are callbacks in Node.js? How do you handle errors with callbacks?
14. Explain Promises in Node.js.
15. What are async/await in Node.js? How do they work?

File System Operations

16. How do you perform file operations in Node.js?
17. Explain the difference between synchronous and asynchronous file operations in Node.js.
18. How do you handle streams in Node.js?

Web Servers and APIs

19. How do you create a simple HTTP server in Node.js?
20. What is Express.js? How do you use it in Node.js applications?
21. Explain middleware in the context of Express.js.
22. How do you handle routing in Express.js?
23. How do you create RESTful APIs using Express.js?

Database Integration

24. How do you connect Node.js with databases like MongoDB or MySQL?
25. What is Mongoose? How do you use it with MongoDB in Node.js?

Error Handling and Debugging

26. How do you handle errors in Node.js applications?
27. What tools and techniques do you use for debugging Node.js applications?

Security

28. What are some common security concerns in Node.js applications?
29. How do you prevent common security vulnerabilities in Node.js applications?

Deployment and Performance

30. How do you deploy Node.js applications? What considerations are important for deployment?

These questions cover foundational and practical aspects of Node.js that are commonly asked in interviews. They are designed to help you prepare effectively for Node.js interviews.

Sure, here's a list of 30 important Express.js interview questions that cover various aspects of the framework:

Core Concepts

1. What is Express.js? Explain its features and advantages.
2. How does Express handle routing?
3. What are middleware functions in Express? How are they used?
4. Explain the difference between `app.get()` and `app.use()` in Express routing.
5. How does error handling work in Express?

Routing and Middleware

6. How do you serve static files in Express?
7. What is the purpose of `app.all()` method in Express routing?
8. How can you handle POST requests in Express?
9. Explain the concept of route parameters in Express with an example.
10. What is the purpose of `next()` function argument in middleware?

Views and Templating

11. What is templating in Express? Which templating engines does Express support?
12. How do you set up and use a template engine like EJS or Handlebars in Express?
13. Explain layout and partials in Express templating engines.

Advanced Topics

14. How do you manage sessions in Express? Explain session storage options.
15. What is Express.js middleware and how does it work?
16. How do you handle file uploads in Express?
17. Explain how to implement authentication and authorization in Express.
18. What are the best practices for structuring an Express application?

Error Handling and Debugging

19. How does Express handle 404 errors?
20. What is the difference between synchronous and asynchronous error handling in Express?
21. How do you debug an Express application?

Performance and Security

22. What are some techniques to improve the performance of an Express application?
23. How can you prevent common security issues in Express applications?

Integration and Deployment

24. How do you deploy an Express application to production?
25. What are the popular deployment options for Express applications?

REST APIs

26. How do you design RESTful APIs using Express?
27. What are the common HTTP methods used in Express for RESTful APIs?
28. Explain how to handle CORS in an Express application.

Testing

29. What are the different approaches to testing an Express application?
30. How do you mock dependencies when testing an Express route or middleware?

These questions cover a broad range of topics in Express.js and should prepare you well for interviews focusing on this framework. Make sure to practice coding examples and scenarios based on these questions to solidify your understanding.
Certainly! Here's a list of 30 important MongoDB interview questions that cover various aspects of the database:

Basics

1. What is MongoDB? How does it differ from traditional SQL databases?
2. Explain the concept of NoSQL databases and where MongoDB fits in.
3. What are the key features of MongoDB?

Data Modeling

4. How does data modeling differ in MongoDB compared to relational databases?
5. What is a document in MongoDB? How is it different from a row in a relational database?
6. Explain embedded documents and arrays in MongoDB with examples.
7. What is a collection in MongoDB? How does it relate to documents?

CRUD Operations

8. How do you insert documents in MongoDB? Provide examples.
9. Explain how to query documents in MongoDB using `find()` and `findOne()`.
10. What is aggregation in MongoDB? Give an example of an aggregation pipeline.

Indexing and Performance

11. What are indexes in MongoDB? Why are they important?
12. How do you create indexes in MongoDB?
13. Explain the different types of indexes available in MongoDB.
14. What are some strategies for improving query performance in MongoDB?

Advanced Querying

15. How do you perform joins in MongoDB?
16. Explain the `$lookup` aggregation stage in MongoDB.
17. What are the differences between `$in` and `$all` operators in MongoDB?

Transactions and ACID Compliance

18. Does MongoDB support transactions? When should you use them?
19. Explain the concept of ACID compliance in the context of MongoDB.

Sharding and Replication

20. What is sharding in MongoDB? How does it help with scalability?
21. How do you set up sharding in MongoDB?
22. Explain MongoDB replication. What are replica sets?

Security

23. How do you secure MongoDB installations?
24. What are the best practices for MongoDB security?

Backup and Recovery

25. How do you back up and restore MongoDB databases?
26. Explain the importance of journaling in MongoDB.

MongoDB Atlas

27. What is MongoDB Atlas? How does it differ from self-hosted MongoDB?
28. What are the benefits of using MongoDB Atlas for cloud-based deployments?

Integration and Tools

29. How do you integrate MongoDB with Node.js applications?
30. What are some useful MongoDB management tools and libraries?

These questions cover a wide range of topics in MongoDB and should help you prepare comprehensively for interviews focusing on this database technology. It's essential to practice querying, indexing, and understanding MongoDB's unique features to feel confident in your responses.

Certainly! Here's a comprehensive list of 50 very important interview questions covering both backend development and databases:

Backend Development

1. What is the difference between frontend and backend development?
2. Explain the concept of a web server and its role in web applications.
3. What are the key features of Node.js?
4. How does Node.js handle asynchronous code?
5. Explain the event-driven architecture of Node.js.
6. What are callbacks in Node.js? How do you handle them?
7. What is npm? How do you use it in Node.js projects?
8. What is Express.js? Explain its main features.
9. How do you handle HTTP requests and responses in Express.js?
10. What are middleware functions in Express.js? Provide examples.
11. Explain the difference between `app.get()` and `app.post()` in Express.js routing.
12. How do you handle query parameters in Express.js?
13. What are RESTful APIs? Explain their characteristics.
14. How do you design RESTful APIs using Express.js?
15. What is authentication? How do you implement it in an Express.js application?
16. Explain JWT (JSON Web Token) and its use in authentication.
17. What is CORS? How do you enable CORS in an Express.js application?
18. How do you handle file uploads in Node.js/Express.js?
19. What is clustering in Node.js? When and why would you use it?
20. How do you implement error handling in Node.js/Express.js applications?
21. What are streams in Node.js? Provide examples of their usage.
22. How would you secure a Node.js application?
23. Explain the concept of middleware in Node.js. How is it different from Express.js middleware?
24. How do you perform unit testing in Node.js applications?
25. What is debugging? How do you debug Node.js applications effectively?

Databases

26. What is a database? Explain the types of databases.
27. What is SQL? How does it differ from NoSQL?
28. What is MongoDB? Explain its key features and use cases.
29. How do you model data in a relational database like MySQL or PostgreSQL?
30. What is normalization and denormalization in databases? When would you use each?
31. What are ACID properties in database transactions? Explain each.
32. How do you optimize SQL queries for performance?
33. Explain different types of SQL joins with examples.
34. What is indexing in databases? Why is it important?
35. How do you create and manage indexes in MongoDB?
36. What are transactions? How do you implement transactions in SQL databases?
37. What is database sharding? When and why would you use it?
38. How does database replication work? What are its benefits?
39. Explain CAP theorem and its implications in database design.
40. What is eventual consistency? How is it achieved in distributed databases?
41. How do you back up and restore databases like MySQL or MongoDB?
42. What are the security best practices for databases?
43. How do you prevent SQL injection attacks?
44. Explain the concept of database triggers. Provide an example.
45. What are stored procedures? When would you use them?
46. How do you handle schema migrations in SQL databases?
47. What are NoSQL databases? When would you choose a NoSQL database over SQL?
48. How do you integrate databases with Node.js applications?
49. What are ORM and ODM? How do they simplify database interactions?
50. What are the advantages and disadvantages of using an ORM/ODM?

These questions cover a broad range of topics in backend development and databases, focusing on both practical application and theoretical understanding. Make sure to practice coding examples and scenarios based on these questions to solidify your understanding and preparation for interviews.