**✅ Basics**

1. **What is Node.js?**  
   Node.js is an open-source, server-side runtime environment that lets you run JavaScript outside the browser.
2. **What is Node.js used for?**  
   It is used to build fast, scalable network applications and APIs, especially real-time web apps.
3. **What are the key features of Node.js?**

* Non-blocking I/O
* Cross-platform

1. **What is the V8 engine?**  
   The V8 engine is Google Chrome's JavaScript engine that compiles JS directly to machine code. Node.js uses it for server-side execution.
2. **Why is Node.js single-threaded?**  
   To handle many connections efficiently using the event loop and non-blocking I/O.
3. **What is the event loop in Node.js?**  
   It continuously checks the call stack and message queue to handle asynchronous tasks without blocking the main thread.
4. **What is non-blocking I/O?**  
   It means Node.js can perform I/O operations without waiting for them to finish, allowing other operations to continue.
5. **What are the advantages of Node.js?**

* Fast and efficient
* Handles many requests simultaneously
* Uses JavaScript for both client and server

**✅ Modules & Packages**

1. **What is require() in Node.js?**  
   require() is used to import built-in, local, or third-party modules.
2. **What are core modules in Node.js?**  
   Pre-built modules like fs, http, path, os, events.
3. **What is npm?**  
   npm stands for Node Package Manager; it installs, updates, and manages project dependencies.
4. **What is package.json?**  
   A file that contains metadata and dependencies of a Node.js project.
5. **How do you install a package locally?**

bash

CopyEdit

npm install package-name

1. **How do you install a package globally?**

bash

CopyEdit

npm install -g package-name

1. **How do you initialize a Node project?**

bash

CopyEdit

npm init

1. **How do you export and import a module in Node.js?**

js

CopyEdit

// export

module.exports = function() {};

// import

const myFunc = require('./myModule');

**✅ Asynchronous Programming**

1. **What is a callback?**  
   A function passed to another function that gets called after the operation finishes.
2. **What is a Promise?**  
   A modern way to handle async code, representing a value that may be available in the future.
3. **What are the states of a Promise?**

* Pending
* Resolved (fulfilled)
* Rejected

1. **What is async/await?**  
   A cleaner way to write promises using async functions and await keyword.
2. **Difference between setImmediate() and process.nextTick()?**

* process.nextTick() executes **before** the next event loop.
* setImmediate() executes **on** the next event loop cycle.

**✅ Other Concepts**

1. **What is the difference between synchronous and asynchronous code?**

* Sync waits for tasks to finish.
* Async doesn’t wait and uses callbacks/promises.

1. **What is the fs module used for?**  
   To perform file system operations like reading/writing files.
2. **How to read a file using fs?**

js

CopyEdit

const fs = require('fs');

fs.readFile('file.txt', 'utf8', (err, data) => {

if (err) throw err;

console.log(data);

});

1. **What is an error-first callback?**  
   A callback pattern where the first argument is err and the second is the result.
2. **What is \_\_dirname?**  
   The absolute path of the current module’s directory.
3. **What is process in Node.js?**  
   It’s a global object that provides info about the current Node process.
4. **How to handle exceptions in Node.js?**  
   Using try...catch, or listening for the 'uncaughtException' event.
5. **What is the use of the events module?**  
   To create and handle custom events in Node.js.
6. **How to create a simple HTTP server in Node.js?**

js

CopyEdit

const http = require('http');

http.createServer((req, res) => {

res.end('Hello, world!');

}).listen(3000);

**🟠 Express.js Interview Questions (20)**

**✅ Basics**

1. **What is Express.js?**  
   Express is a minimal Node.js web framework used to build APIs and web applications.
2. **Why use Express over Node.js alone?**  
   It simplifies routing, request handling, middleware, and improves productivity.
3. **How do you install Express?**

bash

CopyEdit

npm install express

1. **How do you create a basic Express server?**

js

CopyEdit

const express = require('express');

const app = express();

app.listen(3000, () => console.log('Server running'));

**✅ Routing**

1. **How do you define a route in Express?**

js

CopyEdit

app.get('/home', (req, res) => {

res.send('Welcome Home');

});

1. **What is req.params?**  
   Used to capture route parameters (e.g., /user/:id → req.params.id)
2. **What is req.query?**  
   Used to get query string parameters (e.g., /search?name=John → req.query.name)
3. **What is req.body?**  
   Contains data sent via POST (needs middleware like express.json())

**✅ Middleware**

1. **What is middleware in Express?**  
   Functions that run during the request/response cycle. Used for logging, parsing, authentication, etc.
2. **How to use middleware globally?**

js

CopyEdit

app.use((req, res, next) => {

console.log('Request received');

next();

});

1. **What is next() in middleware?**  
   It passes control to the next middleware or route handler.
2. **What are built-in middleware functions in Express?**

* express.json() – parses JSON body
* express.static() – serves static files

1. **How do you serve static files in Express?**

js

CopyEdit

app.use(express.static('public'));

1. **How to handle 404 routes in Express?**

js

CopyEdit

app.use((req, res) => {

res.status(404).send('Page not found');

});

**✅ Request/Response Handling**

1. **How to send a JSON response in Express?**

js

CopyEdit

res.json({ message: 'Success' });

1. **How to set status code in Express?**

js

CopyEdit

res.status(201).send('Created');

1. **What are different types of HTTP methods supported by Express?**  
   GET, POST, PUT, DELETE, PATCH
2. **How to handle POST request data in Express?**

js

CopyEdit

app.use(express.json());

app.post('/data', (req, res) => {

console.log(req.body);

});

**✅ Misc**

1. **What is the difference between app.use() and app.get()?**

* app.use() applies middleware.
* app.get() handles GET requests.

1. **What is the use of routers in Express?**  
   Routers help organize routes in separate files for better structure and maintainability.

If you’d like, I can turn this into a **PDF cheat sheet** or prepare a **mock test** with MCQs or coding questions. Let me know!

4o

Top of Form



Search

Reason

Bottom of Form

ChatGP

**REST API Interview Questions with Answers (for Freshers)**

**✅ 1. What is a REST API?**

**Answer:**  
REST (Representational State Transfer) is an architectural style for designing networked applications. A REST API is an interface that uses HTTP methods to access and manipulate data.

**✅ 2. What are the main HTTP methods used in REST?**

**Answer:**

* GET – Retrieve data
* POST – Create new data
* PUT – Update/replace data
* PATCH – Partially update data
* DELETE – Delete data

**✅ 3. What is the difference between PUT and PATCH?**

**Answer:**

* PUT replaces the entire resource.
* PATCH updates only specific fields of a resource.

**✅ 4. What is a resource in REST?**

**Answer:**  
A resource is any object or data (like a user or product) that can be accessed via a unique URL.

**✅ 5. What are RESTful principles?**

**Answer:**

* Client-Server Architecture
* Statelessness
* Uniform Interface
* Cacheable
* Layered System
* Code on demand (optional)

**✅ 6. What does stateless mean in REST?**

**Answer:**  
Every request is independent; the server does not store any information about previous requests from the client.

**✅ 7. What is the structure of a REST API URL?**

**Answer:**  
Typically:

arduino

CopyEdit

http://domain.com/resource

Example:

nginx

CopyEdit

GET http://api.example.com/users/1

**✅ 8. What is the difference between URI and URL?**

**Answer:**

* **URL** (Uniform Resource Locator) locates a resource.
* **URI** (Uniform Resource Identifier) identifies a resource (can be a URL or name).

**✅ 9. What status codes are commonly used in REST APIs?**

**Answer:**

* 200 OK – Success
* 201 Created – Resource created
* 204 No Content – Success, no data
* 400 Bad Request – Invalid input
* 401 Unauthorized – Not authenticated
* 404 Not Found – Resource not found
* 500 Internal Server Error – Server issue

**✅ 10. What is the purpose of response codes in APIs?**

**Answer:**  
They inform the client about the result of the request (success, error, etc.).

**✅ 11. What is JSON?**

**Answer:**  
JSON (JavaScript Object Notation) is a lightweight format for transferring data between a client and server.

**✅ 12. How do you send data in a POST request?**

**Answer:**  
In the **body** of the request, usually in JSON format.

json

CopyEdit

{

"name": "John",

"email": "john@example.com"

}

**✅ 13. What is the role of headers in REST APIs?**

**Answer:**  
Headers provide metadata, like content type, authentication tokens, etc.

**✅ 14. What is content negotiation?**

**Answer:**  
The server selects the response format (JSON, XML) based on the Accept header sent by the client.

**✅ 15. What are RESTful routes?**

**Answer:**  
Standard URL patterns for CRUD operations.  
Example:

* GET /users – Get all users
* GET /users/:id – Get one user
* POST /users – Create user
* PUT /users/:id – Update user
* DELETE /users/:id – Delete user