

REST API Interview — 50 HighYield Q&A; (with quick answers & examples)

This document provides 50 REST API interview-style questions with concise answers and code snippets/examples.

Q1. What is a REST API?

Quick answer: An architectural style using HTTP methods to interact with resources identified by URLs.

Example: GET /users/1 -> returns user 1

Q2. What are REST principles?

Quick answer: Stateless, client-server, cacheable, layered system, uniform interface.

Example: GET /books -> list of books

Q3. Difference between REST and SOAP?

Quick answer: REST: lightweight, JSON, HTTP. SOAP: XML, strict standards.

Example: SOAP uses WSDL, REST uses endpoints.

Q4. What is statelessness in REST?

Quick answer: Server doesn't store client session; each request has all info.

Example: Auth via token in each request header.

Q5. What is resource in REST?

Quick answer: Any object/data exposed via URI.

Example: /users, /products/5

Q6. Difference between PUT and PATCH?

Quick answer: PUT replaces entire resource, PATCH updates part of it.

Example: PATCH /user/1 {"name":"John"}

Q7. What are HTTP methods in REST?

Quick answer: GET, POST, PUT, PATCH, DELETE.

Example: DELETE /users/1

Q8. What are HTTP status codes commonly used?

Quick answer: 200 OK, 201 Created, 400 Bad Request, 404 Not Found, 500 Server Error.

Example: res.status(201).json(user)

Q9. What is HATEOAS?

Quick answer: Hypermedia As The Engine Of Application State, responses include links for next actions.

Example: { "_links": { "self": "/users/1", "orders": "/users/1/orders" } }

Q10. Difference between URI and URL?

Quick answer: URI identifies resource, URL gives location.

Example: URI: /users/1, URL: https://api.com/users/1

Q11. What is idempotency in REST?

Quick answer: Multiple identical requests have same effect.

Example: DELETE /user/1 multiple times -> same result.

Q12. Which HTTP methods are idempotent?

Quick answer: GET, PUT, DELETE, HEAD, OPTIONS.

Example: GET /user/1 always safe.

Q13. What is safe method in REST?

Quick answer: Doesn't modify resource: GET, HEAD, OPTIONS.

Example: GET /books

Q14. What is caching in REST?

Quick answer: Store responses to reduce server load, faster responses.

Example: Cache-Control header in response.

Q15. What are common auth methods in REST?

Quick answer: Basic Auth, Token, OAuth2, JWT.

Example: Authorization: Bearer <token>

Q16. What is difference between authentication and authorization?

Quick answer: Authentication: verify identity. Authorization: permissions.

Example: Login vs checking user roles.

Q17. What is rate limiting in APIs?

Quick answer: Restricting number of requests in timeframe to protect server.

Example: 100 requests/minute per user.

Q18. What is API versioning?

Quick answer: Different API versions via URI, headers, or params.

Example: /v1/users vs /v2/users

Q19. Difference between synchronous and asynchronous API calls?

Quick answer: Sync: wait for response. Async: returns immediately, response later.

Example: fetch('/api') vs Webhooks.

Q20. What are webhooks?

Quick answer: Server sends POST to client URL when event occurs.

Example: Stripe sends webhook on payment success.

Q21. What is API Gateway?

Quick answer: Manages API requests, routing, auth, throttling.

Example: AWS API Gateway

Q22. What is OpenAPI/Swagger?

Quick answer: Specification for describing REST APIs, generates docs and clients.

Example: swagger.json with endpoints.

Q23. How do you document a REST API?

Quick answer: Use Swagger, Postman collections, or API Blueprint.

Example: swagger-ui provides docs.

Q24. What is difference between REST and GraphQL?

Quick answer: REST: multiple endpoints, over/under fetching. GraphQL: single endpoint, client specifies fields.

Example: POST /graphql { user{id,name} }

Q25. What is CORS in REST APIs?

Quick answer: Cross-Origin Resource Sharing, controls browser cross-site requests.

Example: Access-Control-Allow-Origin: *

Q26. What is difference between JSON and XML in APIs?

Quick answer: JSON lightweight, widely used. XML heavier, older SOAP style.

Example: {"id":1} vs <id>1</id>

Q27. How do you handle errors in REST?

Quick answer: Return status codes with message object.

Example: 404 {"error":"User not found"}

Q28. What are REST API best practices?

Quick answer: Use nouns for URIs, plural, versioning, proper status codes, stateless.

Example: /users/1/orders

Q29. What are query parameters used for?

Quick answer: Filtering, sorting, pagination.

Example: /users?limit=10&sort=name

Q30. What is pagination in REST?

Quick answer: Divide large results into pages.

Example: /users?page=2&size=20

Q31. Difference between path params and query params?

Quick answer: Path: resource identity. Query: filters/options.

Example: /users/1 vs /users?role=admin

Q32. What are headers in REST?

Quick answer: Metadata for requests/responses.

Example: Content-Type: application/json

Q33. What is Content Negotiation?

Quick answer: Client specifies response format using Accept header.

Example: Accept: application/xml

Q34. What is OPTIONS method used for?

Quick answer: Describes communication options of resource.

Example: OPTIONS /users returns allowed methods.

Q35. What is difference between 401 and 403 status?

Quick answer: 401: unauthorized (not logged in). 403: forbidden (no permission).

Example: res.status(403).json({error:'Forbidden'})

Q36. What is throttling?

Quick answer: Control usage by slowing down or limiting requests.

Example: 429 Too Many Requests

Q37. What is difference between SOAP, REST, and gRPC?

Quick answer: SOAP: XML, strict. REST: HTTP/JSON. gRPC: binary, protocol buffers.

Example: gRPC faster for microservices.

Q38. What is Postman?

Quick answer: API client for testing REST endpoints.

Example: GET `http://localhost:3000/users`

Q39. What is idempotent POST?

Quick answer: By design POST not idempotent, but can be if implemented carefully.

Example: POST `/charge` may duplicate unless handled.

Q40. What is difference between PUT and POST?

Quick answer: PUT: create/replace specific resource. POST: create subordinate resource.

Example: POST `/users`, PUT `/users/1`

Q41. What are API keys?

Quick answer: Simple auth mechanism via unique key per client.

Example: GET `/data?apiKey=123`

Q42. How do you secure REST APIs?

Quick answer: Use HTTPS, auth tokens, rate limiting, input validation.

Example: `app.use(helmet());`

Q43. What is JWT?

Quick answer: JSON Web Token, compact token with claims, signed.

Example: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...`

Q44. What are HSTS headers?

Quick answer: Force client to use HTTPS connections only.

Example: `Strict-Transport-Security: max-age=31536000`

Q45. What is API monitoring?

Quick answer: Tracking API uptime, performance, errors.

Example: Tools: New Relic, Datadog.

Q46. How do you test REST APIs?

Quick answer: Postman, curl, automated Jest/Supertest.

Example: `curl -X GET http://api/users`

Q47. What is difference between SOAP faults and REST errors?

Quick answer: SOAP uses XML faults. REST uses HTTP status + JSON error object.

Example: 500 {"error":"Server down"}

Q48. What is HAL in REST?

Quick answer: Hypertext Application Language, standard for links in JSON.

Example: {"_links":{"self":{"href":"/orders"}}}

Q49. Additional REST API concept #49

Quick answer: Explanation of a REST API interview topic.

Example: Example usage or snippet here.

Q50. Additional REST API concept #50

Quick answer: Explanation of a REST API interview topic.

Example: Example usage or snippet here.