**1. What is Postman?**  
**Answer:**  
*Sir, I would like to explain that Postman is a powerful API testing tool that allows us to send HTTP requests to APIs and view their responses. It's widely used by developers for testing, debugging, and documenting APIs. It supports REST, SOAP, and GraphQL as well.*

**2. Why do developers use Postman?**  
**Answer:**  
*Sir, I would like to suggest that developers use Postman because it provides a very user-friendly interface to test APIs quickly without writing any code. It helps in checking the request-response structure, status codes, performance, and can also automate testing using collections and scripts.*

**3. What are the common HTTP methods you use in Postman?**  
**Answer:**  
*Sir, I would like to explain that the most commonly used HTTP methods in Postman are GET, POST, PUT, PATCH, and DELETE. Each method is used for a specific purpose — for example, GET is used to fetch data, POST to create data, PUT/PATCH to update data, and DELETE to remove data.*

**4. How do you send a POST request using Postman?**  
**Answer:**  
*Sir, to send a POST request in Postman, I select POST from the dropdown menu, then enter the API URL. In the “Body” tab, I choose ‘raw’ and select ‘JSON’ format, then I add the JSON payload I want to send. Finally, I click on 'Send' to make the request.*

**5. What is the difference between PUT and PATCH in Postman?**  
**Answer:**  
*Sir, I would like to explain that both PUT and PATCH are used to update data. The main difference is — PUT replaces the entire resource with the new one, while PATCH only updates the specific fields we mention. For example, if I only want to change the name, PATCH is more efficient.*

**6. What is the use of the Pre-request Script tab in Postman?**  
**Answer:**  
*Sir, I would like to suggest that the Pre-request Script tab in Postman is used to write JavaScript code that runs* ***before*** *the request is sent. It’s helpful for setting dynamic values like tokens, timestamps, or generating random data which can then be used in the request.*

**7. How do you test an API in Postman?**  
**Answer:**  
*Sir, I would like to explain that to test an API in Postman, I enter the API endpoint, choose the right HTTP method, set headers if needed, and send the request. Then I analyze the response body, status code, and write test scripts to validate the API behavior using JavaScript.*

**8. Can you save requests in Postman? If yes, how?**  
**Answer:**  
*Yes sir, Postman allows us to save requests. After entering the request details, I click on the “Save” button, name the request, and save it inside a Collection. Collections help keep related requests organized and reusable for future testing or automation.*

**9. What is a Collection in Postman?**  
**Answer:**  
*Sir, I would like to explain that a Collection in Postman is a group of saved API requests. It helps to organize multiple endpoints together for a particular project or module. Collections also support features like tests, variables, documentation, and automation with Collection Runner.*

**10. How do you organize APIs in Postman?**  
**Answer:**  
*Sir, I organize APIs in Postman using Collections and Folders. Each module or feature can have its own folder inside a Collection. I use naming conventions and environment variables to make requests clean, reusable, and easy to navigate.*

Let me know if you want me to continue with Q11–Q20 in the same style, or convert all into a printable PDF/Notion copy.

**You said:**

next 10

**ChatGPT said:**

Absolutely Bhai! Here's the **next 10 Postman interview questions (Q11–Q20)** with smart, **interview-style answers**, starting with **"Sir, I would like to explain/suggest..."** so you can confidently speak like a pro:

**✅ Postman Interview Answers (Q11 to Q20)**

**11. What is an environment in Postman?**  
**Answer:**  
*Sir, I would like to explain that an environment in Postman is a set of key-value pairs, like variables, which help manage different configurations such as development, staging, and production. We can use {{variable\_name}} in requests, and Postman will replace them with actual values from the environment.*

**12. What are global and environment variables in Postman?**  
**Answer:**  
*Sir, I would like to suggest that environment variables are specific to a selected environment, while global variables can be accessed from anywhere across all environments. I use environment variables for project-specific setups and global ones for values used across multiple collections.*

**13. How do you create and use variables in Postman?**  
**Answer:**  
*Sir, to create a variable, I go to the Environment section or Global section, and define a key-value pair. Then in the request, I use it like {{myVariable}}. Postman automatically replaces it with the actual value when sending the request.*

**14. What is the Collection Runner in Postman?**  
**Answer:**  
*Sir, I would like to explain that the Collection Runner is used to run a collection of requests in a sequence. It’s helpful for testing workflows, running data-driven tests using CSV/JSON files, and validating the entire API flow automatically.*

**15. What is the difference between Params and Body in Postman?**  
**Answer:**  
*Sir, I would like to explain that ‘Params’ are used for query parameters in GET requests, and they appear in the URL. The ‘Body’ tab is used to send data in POST, PUT, or PATCH requests. It can be in form-data, raw JSON, or other formats.*

**16. What are Tests in Postman? How do you write one?**  
**Answer:**  
*Sir, I would like to suggest that Tests in Postman are small JavaScript scripts written in the “Tests” tab to validate response data. For example, I can write pm.test("Status is 200", () => pm.response.to.have.status(200)); to check if the status code is 200.*

**17. Can Postman be used for automation?**  
**Answer:**  
*Yes Sir, Postman supports automation using Collection Runner and Newman CLI. I can automate test cases, run them in bulk with data files, and even schedule runs through CI/CD tools like Jenkins or GitHub Actions.*

**18. What is Newman in Postman?**  
**Answer:**  
*Sir, I would like to explain that Newman is the command-line companion of Postman. It allows us to run Postman collections in a terminal or through scripts. This is especially useful for integrating Postman tests into build pipelines.*

**19. How can you test APIs in Postman using test scripts?**  
**Answer:**  
*Sir, I use the “Tests” tab in Postman to write JavaScript-based test scripts. I validate things like response status code, headers, body content, or even custom logic. Postman provides the pm API to make it simple and powerful for test validation.*

**20. Can you write a test in Postman to validate response status is 200?**  
**Answer:**  
*Yes Sir, the test would be:*

javascript

CopyEdit

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

*This checks that the response returned a 200 OK status. I can also add multiple such tests for headers, response time, and body values.*

Let me know if you want **Q21–Q30 next**, or if you'd like a **clean single file** with all 50 questions + answers together. Ready when you are, Bhai! 💪

**You said:**

next 10

**ChatGPT said:**

Bhai, here are the **next 10 Postman interview questions (Q21–Q30)** with smart, **interview-style answers** that start like you’re explaining them to an interviewer — professional yet confident:

**✅ Postman Interview Answers (Q21 to Q30)**

**21. Can you explain different types of authorization in Postman?**  
**Answer:**  
*Sir, I would like to explain that Postman supports multiple authorization types such as Basic Auth, Bearer Token, API Key, OAuth 1.0, OAuth 2.0, and more. I usually choose the one required by the API. For example, I use Bearer Token for JWT-based APIs and OAuth2 for more secure APIs like Google APIs.*

**22. How do you pass a token in Postman?**  
**Answer:**  
*Sir, I copy the token from the login response or environment variable and go to the “Authorization” tab. I select “Bearer Token” and paste the token. Alternatively, I can manually add it to the Headers tab like: Authorization: Bearer {{token}}.*

**23. What is the difference between form-data and x-www-form-urlencoded in Postman?**  
**Answer:**  
*Sir, I would like to explain that form-data is used to upload files and send complex forms, especially for multipart/form-data. x-www-form-urlencoded sends key-value pairs as URL-encoded text. It’s mostly used in HTML form submissions.*

**24. What are Pre-request Scripts in Postman?**  
**Answer:**  
*Sir, Pre-request Scripts are written in JavaScript and executed before the request is sent. I use them to set or modify environment variables, create timestamps, or generate authentication signatures. For example, I generate a token dynamically in the Pre-request Script.*

**25. Can Postman be integrated with CI/CD?**  
**Answer:**  
*Yes Sir, Postman collections can be run using Newman in a CI/CD pipeline like Jenkins, GitHub Actions, or GitLab CI. I export the collection and environment file and then use a script like newman run collection.json -e environment.json to automate testing.*

**26. How can you handle dynamic values like timestamps or UUIDs in Postman?**  
**Answer:**  
*Sir, I use JavaScript inside Pre-request Scripts to generate dynamic values. For example, new Date().toISOString() gives me the current timestamp, and for UUIDs I use libraries or Postman’s built-in method pm.variables.replaceIn('{{$guid}}').*

**27. What is the difference between collection-level and request-level variables or scripts?**  
**Answer:**  
*Sir, collection-level scripts and variables apply to all requests in that collection. Request-level ones apply only to a specific request. I use collection-level for common logic like setting tokens, and request-level for specific API scenarios.*

**28. How do you import/export Postman collections?**  
**Answer:**  
*Sir, to export a collection, I click on the three dots beside the collection and choose “Export.” To import, I use the “Import” button at the top left and upload a collection file or paste the URL if it's from a GitHub or shared workspace.*

**29. What is Postman Monitor?**  
**Answer:**  
*Sir, Postman Monitor allows me to run a collection at scheduled intervals. It helps in monitoring API performance and uptime. I get alerts if any test fails. I use it to ensure that critical endpoints are working 24/7.*

**30. How do you validate JSON response in Postman?**  
**Answer:**  
*Sir, I use the pm.response.json() function to parse the response, then use assertions to check values. For example:*

javascript

CopyEdit

let res = pm.response.json();

pm.test("Username is correct", function () {

pm.expect(res.username).to.eql("bhushan");

});

*This helps in verifying specific fields in a structured JSON response.*

Let me know when you're ready for **Q31–Q40** or if you want a **PDF/Notepad file with all 50 Q&A** in one copy. Always ready for you, Bhai 💻🔥

**You said:**

next 10

**ChatGPT said:**

Here are the **next 10 Postman interview questions (Q31–Q40)** with **smart, interview-style answers**—just how you’d confidently explain them to an interviewer:

**✅ Postman Interview Answers (Q31 to Q40)**

**31. How do you test file uploads in Postman?**  
**Answer:**  
*Sir, to test file uploads, I select the method as POST, go to the Body tab, choose form-data, and then add a key with type set to “File.” I browse and select the file from my system. It mimics how a frontend would upload files to the server.*

**32. How do you test APIs with different environments like dev, staging, and prod?**  
**Answer:**  
*Sir, I create separate environments in Postman like dev, staging, and production. Each has variables like base\_url, token, etc. Then I use {{base\_url}} in requests, so I can switch environments easily without changing the request code.*

**33. What is chaining requests in Postman?**  
**Answer:**  
*Sir, chaining means using the output of one request as input for another. I achieve this by saving response data (like tokens or IDs) into environment or global variables in the Tests tab, and then using those variables in subsequent requests.*

**34. How do you debug issues in Postman requests?**  
**Answer:**  
*Sir, I usually check the response status code, headers, and body. If the issue is unclear, I use console.log() in scripts and view it in the Postman Console. It helps in inspecting variables, responses, and errors during request execution.*

**35. What is the difference between Global and Environment variables?**  
**Answer:**  
*Sir, Global variables are accessible across all workspaces and environments. Environment variables are limited to the selected environment. I prefer using environment variables to keep my test data organized and context-specific.*

**36. Can Postman be used for performance testing?**  
**Answer:**  
*Sir, Postman isn’t ideal for full performance testing, but it can do basic response time checks. For more advanced load testing, I export the collection and use Newman with tools like Artillery or integrate with JMeter.*

**37. What are test scripts in Postman?**  
**Answer:**  
*Sir, test scripts are written in JavaScript inside the Tests tab. I use them to validate response codes, check specific fields in JSON, or even set variables. For example: pm.test("Status code is 200", () => pm.response.to.have.status(200));.*

**38. How do you handle pagination in Postman APIs?**  
**Answer:**  
*Sir, for paginated APIs, I extract the next page URL or page number from the response, store it in a variable, and use it in the next request. I can also use Pre-request Scripts and loop logic with Postman’s Collection Runner or external tools like Newman.*

**39. How do you test negative scenarios in Postman?**  
**Answer:**  
*Sir, I test APIs with missing headers, invalid input data, wrong endpoints, and incorrect tokens to ensure the API responds with proper error codes like 400, 401, or 404. These negative cases help me ensure the API is robust and handles failures correctly.*

**40. Can you explain what is Postman Collection Runner?**  
**Answer:**  
*Sir, Collection Runner lets me run all or selected requests in a collection in sequence. I can run it with data files like CSV or JSON for data-driven testing. It’s very useful for automation, regression testing, and bulk request execution.*

Let me know when you want the final **Q41–Q50** or if you'd like the **entire set as a PDF or DOCX file** ready to download. Always here to back you up, Bhai 💪📘

**You said:**

next 10

**ChatGPT said:**

Here are the final **10 Postman interview questions (Q41–Q50)** along with **smart, interview-style answers** just the way you’d speak to an interviewer confidently.

**✅ Postman Interview Answers (Q41 to Q50)**

**41. How do you import and export collections in Postman?**  
**Answer:**  
*Sir, to export, I click on the collection, select "Export", and save it as a JSON file. To import, I use the “Import” button and upload the collection JSON file. It’s very helpful for sharing APIs with teams or migrating across machines.*

**42. What is a Postman Monitor?**  
**Answer:**  
*Sir, Postman Monitor lets me schedule and automate collection runs. For example, I can monitor an API every hour and receive alerts if it fails. It’s useful for uptime checks and regression tests in production environments.*

**43. Can you run Postman collections via CI/CD pipelines?**  
**Answer:**  
*Yes sir, I use Newman, Postman’s command-line tool. I integrate it with CI tools like GitHub Actions, Jenkins, or GitLab CI. This helps me automate API tests during build or deployment stages.*

**44. What’s the role of Postman Pre-request scripts?**  
**Answer:**  
*Sir, Pre-request scripts run before the API request is sent. I use them to set timestamps, auth tokens, or manipulate variables. It’s like preparing everything before hitting the API.*

**45. What are the different authentication methods Postman supports?**  
**Answer:**  
*Sir, Postman supports Basic Auth, Bearer Token, API Key, OAuth 1.0, OAuth 2.0, Digest Auth, Hawk Auth, and AWS Signature. I usually use OAuth 2.0 and Bearer Token for secured APIs.*

**46. What’s the difference between pm.environment.set() and pm.globals.set()?**  
**Answer:**  
*Sir, pm.environment.set() sets variables for the selected environment, while pm.globals.set() sets global variables accessible everywhere. I use environment variables for test-specific data and global ones for values reused across collections.*

**47. How do you pass dynamic values in Postman requests?**  
**Answer:**  
*Sir, I use variables like {{token}} in headers or body. In scripts, I dynamically generate values using JavaScript, like random emails or timestamps, and set them using pm.environment.set() before sending requests.*

**48. How do you test APIs that use JWT tokens?**  
**Answer:**  
*Sir, I first authenticate using the login API to receive the JWT token. Then I save it to a variable using pm.environment.set("token", value). After that, I use {{token}} in the Authorization header for protected requests.*

**49. How do you ensure API security testing in Postman?**  
**Answer:**  
*Sir, I perform tests like SQL injection attempts, missing token checks, role-based access violations, and expired token usage. These help me ensure the API doesn’t expose any sensitive or restricted data.*

**50. Why do companies prefer Postman for API testing?**  
**Answer:**  
*Sir, companies love Postman because it’s user-friendly, supports automation, environment management, scripting, test validation, and integrates well with CI/CD. It reduces manual effort and ensures consistent API quality.*