

Postman Interview Questions – Part 2

 Advanced Topics • Real-World Scenarios • Automation Focused

1. How do you test file upload functionality in Postman?

Use **form-data** body type → select file key → choose file from system.

2. How do you send nested JSON data in Postman?

Select **raw** → **JSON** in Body tab and send:

```
{
  "user": {
    "name": "Parag",
    "email": "test@example.com"
  }
}
```

3. What is the difference between PUT and PATCH?

- **PUT** replaces the whole resource.
 - **PATCH** updates only specific fields.
-

4. How do you validate response time in Postman?

```
pm.test("Response time < 500ms", () => {
  pm.expect(pm.response.responseTime).to.be.below(500);
});
```

5. How do you store and reuse response values?

```
pm.environment.set("userId", pm.response.json().id);
```

Then use `{{userId}}` in next requests.

6. What is Collection Runner in Postman?

It allows running an entire set of requests (collection) with test data in iterations.

7. What is the use of Pre-request Script in Postman?

Used to run JavaScript **before** the request — often for generating tokens, timestamps, or setting variables.

8. How do you use dynamic variables in Postman?

Use:

```
{{randomEmail}}, {{randomInt}}, {{guid}}, {{timestamp}}
```

9. What are environments in Postman?

Environments allow using different variable sets like base URL, auth tokens — useful for switching between dev/stage/prod.

10. What is Newman in Postman ecosystem?

Newman is the command-line tool to run Postman collections in CI/CD pipelines or scripts.

11. How can you import test data in Postman?

Use **Collection Runner** with CSV or JSON files for data-driven testing.

12. How do you run Postman tests in Jenkins?

Install Newman in Jenkins and use CLI:

```
newman run collection.json -e env.json
```

13. How can you validate a specific JSON field?

```
pm.test("Validate username", () => {  
  pm.expect(pm.response.json().username).to.eql("parag123");  
});
```

14. What are global vs environment variables?

- **Global:** Available across all workspaces.
 - **Environment:** Scoped to specific collections/sessions.
-

15. What's the Postman Console used for?

To debug requests, log data (`console.log()`), and inspect errors during request execution.

16. Can you write conditional tests in Postman?

```
if (pm.response.code === 200) {  
  pm.test("Success", () => true);  
}
```

17. How do you handle authentication in Postman?

Use the **Authorization** tab to configure:
Basic Auth, Bearer Token, OAuth 2.0, or API key.

18. What are monitors in Postman?

They run collections on schedule (like uptime checks or health checks) and notify on failures.

19. How do you handle pagination while testing APIs?

Manually update query params like `?page=2` or use a loop in test scripts to fetch paginated results.

20. Can Postman be used for automated testing?

Yes, by combining **Collections + Tests + Newman + CI tools** like Jenkins or GitHub Actions.