

✅ Day 4: Tests & Scripts – Automate Validation

👤 Created by: Deep Ghinaiya

💼 Software Developer

🔗 linkedin.com/in/deep-ghinaiya

📌 What You'll Learn:

- What are **tests** in Postman
- How to write simple **JavaScript test scripts**
- Use built-in Postman **snippets**
- Validate **status codes**, **response time**, and **body content**
- Chain requests using **variables and tests**

🧪 What Are Tests?

Postman tests are small JavaScript scripts that run **after a request** completes. They help you **automate checks** and make sure your API behaves as expected.

🔧 Simple Test Examples:

✅ Check if status is 200:

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});
```

✅ Check response time:

```
pm.test("Response time is less than 500ms", function () {  
    pm.expect(pm.response.responseTime).to.be.below(500);  
});
```

✅ **Validate JSON body value:**

```
pm.test("Title is present", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData.title).to.not.be.undefined;  
});
```

🧩 **Built-in Snippets:**

Postman provides test snippets like:

- Status code check
- Response time check
- Content-type validation
- JSON value check

You can insert them quickly from the **Tests** tab on any request.

🔗 **Bonus – Set Variable from Response:**


You can extract data from one response and save it for the next request:

```
let jsonData = pm.response.json();  
pm.environment.set("post_id", jsonData.id);
```



This is super useful for **chaining requests**.

🧠 **Pro Tips:**

- ✅ Keep tests short & specific
- ✅ Use pm.expect() for clean syntax
- ✅ Combine with Collection Runner for automation
- ✅ Write reusable scripts across environments

 Full Series:

 github.com/Ghinaiya-Deep/Skill-Stack-by-DeepLogic-Labs

 Found this helpful? Star  the repo & share it with your team!