



VERIFICATIONS & ASSERTIONS WITH POSTMAN



WHAT TO EXPECT

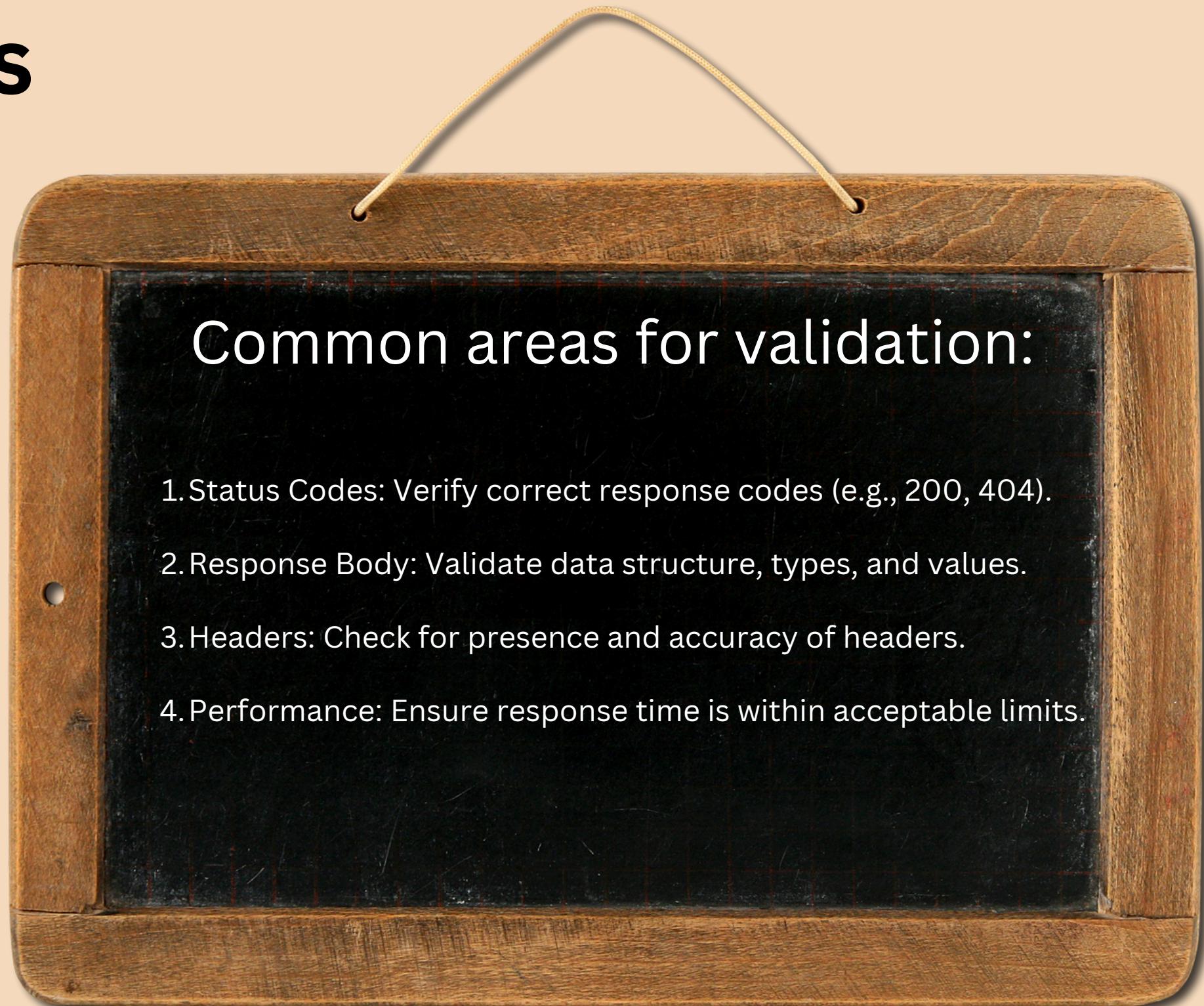
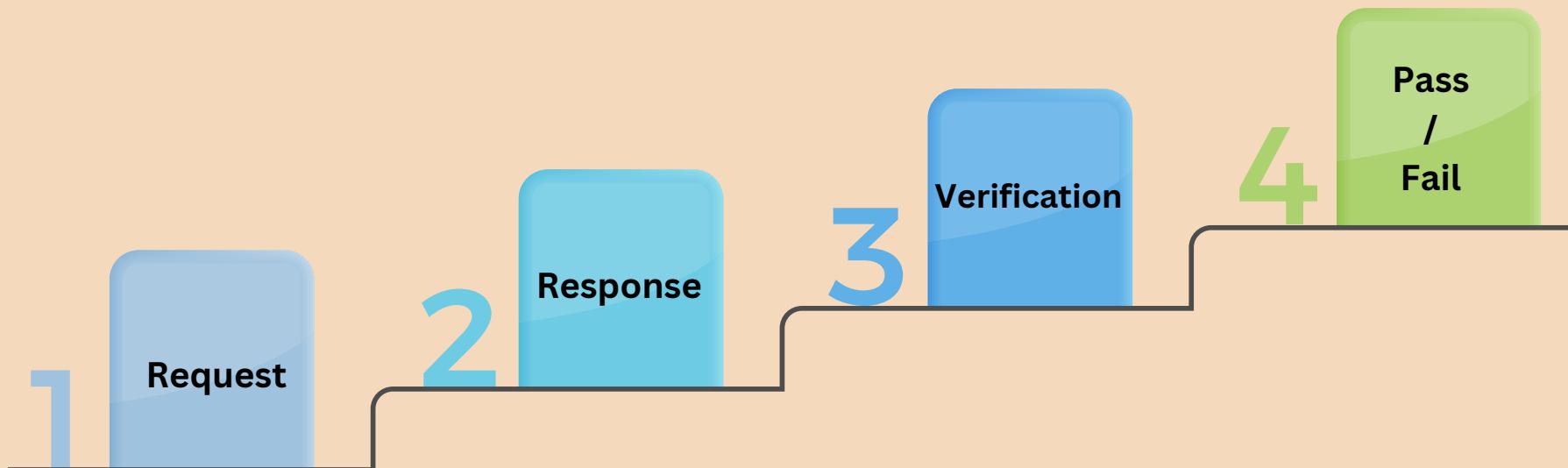
“let’s”
DO THIS

- 1 Verifications & Assertions
- 2 Using Chai Assertions in Postman
- 3 Several types of Assertions
- 4 Saving and Reusing Data
- 5 Postman vs. SoapUI for Verifications

Verifications & Assertions



- **Verifications** ensure API responses meet **expectations**.
- **Assertions** are specific checks within **verifications**.



1. Status Codes: Verify correct response codes (e.g., 200, 404).
2. Response Body: Validate data structure, types, and values.
3. Headers: Check for presence and accuracy of headers.
4. Performance: Ensure response time is within acceptable limits.



Using Chai Assertions in Postman

Postman uses the Chai Assertion Library for its testing framework.



Benefits :

- Human-readable syntax for easy test creation.
- Supports BDD (Behavior-Driven Development) style testing.
- Extensive support for complex validations.

The screenshot shows the Postman application interface. In the center, there is a test script editor window with the following content:

```
1 pm.test("Status code is 200", function() {
2     pm.response.to.have.status(200);
3 })
```

Below the script, the "Test Results" tab is selected, showing a single test result:

All	Passed	Skipped	Failed
PASS	Status code is 200		

On the right side of the interface, there is a sidebar titled "SNIPPETS" containing various assertion snippets:

- Test scripts are written in JavaScript, and are run after the response is received.
- Learn more about tests
- SNIPPETS
- Response body: Contains string
- Response body: Convert XML body to a JSON Object
- Response body: Is equal to a string
- Response body: JSON value check
- Response headers: Content-Type header check
- Response time is less than 200ms
- Send a request

Several Types of Assertions



EXAMPLE

Status Code	pm.response.to.have.status(200);
Response Body Structure	pm.expect(pm.response.json()).to.have.property('id');
Headers	pm.expect(pm.response.headers.get('Content-Type')).to.include('application/json');
Data Validations	pm.expect(pm.response.json().userId).to.equal(1);
Performance	pm.expect(pm.response.responseTime).to.be.below(500);

Saving and Reusing Data

- Extract data from one response and reuse it in subsequent requests.
- Enables dynamic testing scenarios.

```
let responseData = pm.response.json();
pm.environment.set("userId", responseData.userId);
```

Request 1

```
let userId = pm.environment.get("userId");
pm.expect(userId).to.be.a('number');
```

Request 2

Postman vs. SoapUI for Verifications

Feature	Postman	SoapUI
Ease of Use	Sure !	Sure !
Automation	Supports scripting and chaining	Supports scripting and chaining
Assertion Language	Chai Assertion Library	Groovy scripting & Javascript
Collaboration	Sure !	Sure !
Learning Resources	Plenty	Plenty



LIGHTS
CAMERA
ACTION





“

See you in Next Video

