

Foundations of Software Testing

2.3 Advanced Postman Features

Learning objectives

- ▶ Why organize your Postman work
- ▶ Using Collections to group requests
- ▶ Defining Variables and Environments
- ▶ Adding Tests and Pre-request Scripts
- ▶ Exporting and sharing your work

Why Go Beyond Basic Requests

- ▶ Manual testing can become repetitive
- ▶ Reusing URLs or headers saves time
- ▶ Structured organization supports team work
- ▶ Scripts and variables improve consistency
- ▶ Prepares you for automated test integration later

What Are Collections?

- ▶ A Collection is a folder for API requests
- ▶ Helps organize requests by:
 - ▶ Project
 - ▶ Feature
 - ▶ Resource type
- ▶ You can run all requests in a collection sequentially
- ▶ Supports sharing and automation later

Creating a Collection

- ▶ Click Collections in Postman sidebar
- ▶ Click + New Collection
- ▶ Name it (e.g., "JSONPlaceholder Tests")
- ▶ Save each request (GET, POST, etc.) into the collection
- ▶ Optionally, add descriptions for each request

What Are Variables?

- ▶ Variables store reusable values
- ▶ Avoids hardcoding things like URLs or tokens
- ▶ Types:
 - ▶ Global: available everywhere
 - ▶ Collection: specific to one collection
 - ▶ Environment: changes with environment (dev, test, prod)
- ▶ Use syntax: {{variableName}}

What Are Environments?

- ▶ Define sets of variables for different setups
 - ▶ baseUrl = local / test / production
- ▶ Switch between environments with one click
- ▶ Great for testing same API in multiple contexts
- ▶ Each environment stores its own values

Demo: Variables and Environments

- ▶ Demo Goals:
 - ▶ 1. Create a variable `baseUrl = https://jsonplaceholder.typicode.com`
 - ▶ 2. Use it in a request: `{{baseUrl}}/posts/1`
 - ▶ 3. Duplicate the environment and change `baseUrl` (for example to another API)
 - ▶ 4. Show how switching environment changes request behavior
 - ▶ 5. Discuss how this helps manage multiple setups easily

Lab: Create a Structured Postman Collection

- ▶ Goal: Organize API requests and add simple validations.
- ▶ Instructions:
 - ▶ 1. Open Postman → Create a new Collection called "My Public API Tests"
 - ▶ 2. Add a variable: baseUrl = <https://jsonplaceholder.typicode.com>
- ▶ 3. Add three requests:
 - ▶ GET {{baseUrl}}/posts
 - ▶ GET {{baseUrl}}/posts/1
 - ▶ POST {{baseUrl}}/posts (with JSON body)
- ▶ Run each request and observe test results.

Lab: Optional Challenge: Parameterized Requests

- ▶ If time allows:
 - ▶ Add a new variable: `{{postId}} = 1`
 - ▶ Use it in `GET {{baseUrl}}/posts/{{postId}}`
 - ▶ Change the variable value and re-run
 - ▶ Observe how it changes results dynamically

Lab: Reflection

- ▶ How did variables make testing easier?
- ▶ What's the benefit of organizing requests into collections?
- ▶ How do simple test scripts add confidence to manual testing?
- ▶ What challenges did you encounter?

Key Takeaways

- ▶ Collections organize related requests
- ▶ Variables and environments improve reusability
- ▶ You can now manage structured, repeatable API testing in Postman