Foundations of Software Testing

2.2 Introduction to Manual API Testing with Postman

Learning objectives

- What is Postman?
- Why use it for API testing
- The Postman interface
- Sending basic requests (GET, POST, PUT, DELETE)
- Viewing and understanding responses

What Is Postman?

- A tool for exploring and testing APIs
- Sends HTTP requests and displays responses
- Works with any web API – public or private
- Lets you experiment without writing code
- Used by developers, testers, and DevOps teams
- Available as desktop app or web version

Why Use Postman for API Testing

- Makes it easy to understand request–response flow
- Helps visualize HTTP methods and status codes
- Ideal for testing during development
- Provides immediate feedback from the API
- Foundation for learning automated testing later

The Postman Interface Overview

- Main Components:
- Request tab: Enter API URL and choose HTTP method
- Headers tab: Add metadata like Content-Type
- Body tab: Define data for POST/PUT requests
- Send button: Executes the request
- Response panel: Displays status, headers, and data

Making Your First Request

- 1. Open Postman
- 2. Choose method: GET
- 3. Enter URL: https://jsonplaceholder.typicode.com/posts/1
- 4. Click Send
- 5. Observe:
  - Response body (JSON)
  - Status 200 OK
  - Headers (Content-Type, Date, etc.)

Understanding the Response

Part	Description	Example
Status Code	Indicates result	200 OK
Headers	Metadata about response	Content-Type: application/json
Body	Data returned	JSON object representing a post

Exploring Common HTTP Methods

Method	What It Does	Example in Postman
GET	Retrieve data	/posts/1
POST	Create data	/posts + JSON body
PUT	Replace data	/posts/1 + JSON body
DELETE	Remove data	/posts/1

Demo: Testing an API Manually with Postman

- API: https://jsonplaceholder.typicode.com
- Show these actions:
  - 1. GET /users → returns list of users (200 OK)
  - 2. GET /users/1 → single user (200 OK)
  - 3. GET /users/9999 → 404 Not Found
  - 4. POST /posts with JSON body → 201 Created
- Discuss:
  - Status codes
  - Response body (JSON)
  - How Postman visually separates request & response

Lab: Manual API Testing with Postman

- 1. Open Postman.
- 2. Create a new request tab.
- 3. Try these requests against https://jsonplaceholder.typicode.com:
  - GET /posts/1
  - GET /users/1
  - GET /comments?postId=1
  - POST /posts with Body → raw JSON:
    - {"title": "API Testing with Postman","body": "Learning manual testing is fun!","userId": 1}
- 4. Observe the status code, headers, and body for each.

Lab: Expected Results

Request	Expected Status	Notes
GET /posts/1	200 OK	Returns JSON object with post data
GET /users/1	200 OK	Returns user object
GET /comments?postId=1	200 OK	Returns array of comments
POST /posts	201 Created	Returns JSON with new post id

Key Takeaways

- Postman is a powerful tool for exploring and testing APIs
- You can send any HTTP request and inspect the response
- Manual testing helps you understand API behavior
- Status codes and response bodies tell the story of each call
- Foundation for automation and validation tests later with JUnit and Rest Assured