



KONGU ENGINEERING COLLEGE
(Autonomous)



Perundurai, Erode – 638 060

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

**22CSE03 – BUILDING ENTERPRISE
APPLICATION**

**WEB SERVICE AUTOMATION TESTING
USING POSTMAN**

Submitted By:

Mathav Ra (22CSR117)

Overview of Web Service Automation with Postman:

What is Web Service Automation?

Web service automation is the process of using automated tools to test APIs, ensuring they function as expected. This approach validates the interactions between different software components and helps catch issues early in the development cycle.

Why Use Postman for Web Service Testing?

Postman is a powerful, user-friendly tool for web service testing and automation. It allows developers and testers to create, manage, and automate requests to RESTful APIs, covering various testing scenarios, including:

- **Functional Testing:** Verifying that each endpoint works as expected.
- **Performance Testing:** Checking API speed and response time.
- **Security Testing:** Validating authentication, authorization, and handling of sensitive data.

Key Benefits of API Automation with Postman:

1. **Efficiency:** Reduces the time spent on manual testing by allowing tests to run automatically.
2. **Reliability:** Ensures consistent results across multiple test runs.
3. **Scalability:** Allows tests to be created and scaled easily for large numbers of test cases.
4. **Easy Debugging:** Postman provides detailed logs and tools to help debug issues within APIs.

5. **Integration with CI/CD Pipelines:** Postman collections can be integrated with Continuous Integration/Continuous Deployment (CI/CD) systems, enabling automated testing on code change.

Password Management API Description:

This Password Management API is designed to enable users to change their password securely. It verifies the user's current password, validates the new password against security rules, and updates it if all conditions are met.

Base URL: <http://<your-domain-or-localhost>:<port>/>

Endpoint:

- Endpoint: /change-password
- HTTP Method: POST

Request Parameters:

Parameter	Type
Email	String
Current Password	String
New Password	String

Password Requirements:

1. Minimum Length: 8 characters.
2. Upper and Lower Case: Must contain both uppercase and lowercase letters.

3. Special Character: Must contain at least one special character from !@#\$\$%^&*.
4. Numeric Digit: Must include at least one number.

Sample Request:

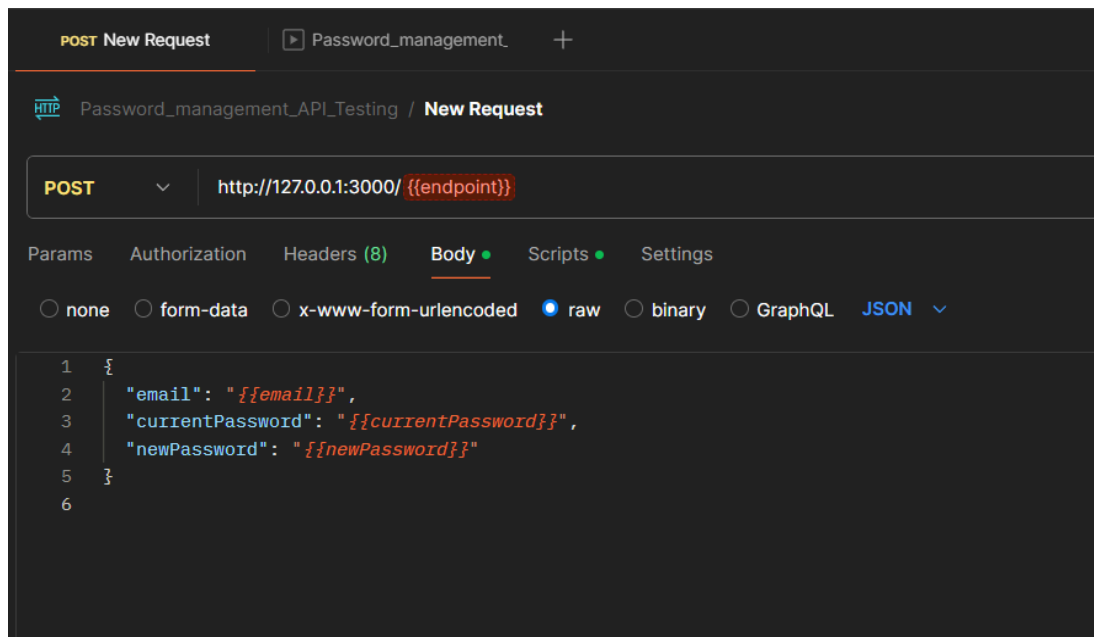


Figure 1.1: Request and endpoint url in Postman

Sample Responses:

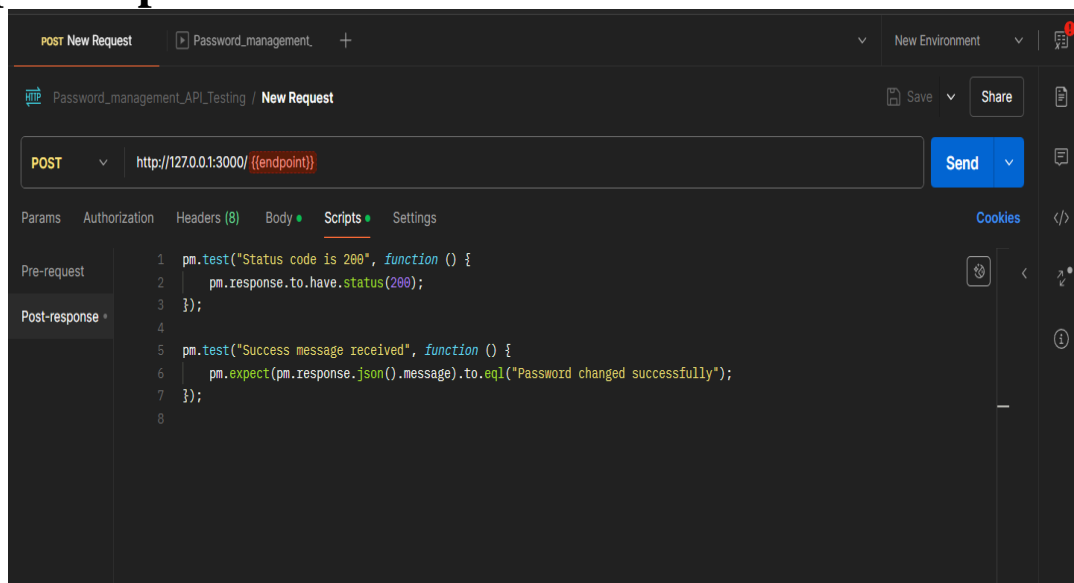


Figure 1.2: Test scripts for input data

Test Case (XML):

- Create a CSV file (e.g.,password_testcase.csv) with your test data.
- Column name are endpoint, email, currentPassword, newPassword.

	A	B	C	D	E
1	endpoint	email	currentPassword	newPassword	expected_status
2	change-password	alice@example.com	Password123!	NewPassword123!	200
3	change-password	alice@example.com	WrongPassword	NewPassword123!	400
4	change-password	bob@example.com	Password123!	short!	400
5	change-password	bob@example.com	Password123!	NoDigitPassword!	400
6	change-password	bob@example.com	Password123!	Noupper123!	400
7	change-password	bob@example.com	Password123!	NOLOWER123!	400
8	change-password	nonexistent@example.com	Password123!	ValidPassword123!	404
9	change-password	mathav@example.com	Mathav@123	[new]	404
10	change-password	mathav@example.com	Password123!	NewPassword123!	200
11	change-password	adi@example.com	adi@12345678	Password124	404
12	change-password	bob@example.com	Password123!	NewPassword123@	200
13					

Figure 1.3: password_testcaseTest.csv

Usage Scenarios:

1. **Correct Request:** When the user provides valid email, current password, and a new password that meets all requirements, they will receive a success message.
2. **Invalid Current Password:** If the provided current Password is incorrect, a 401 Unauthorized error is returned.
3. **Invalid New Password:** If the newPassword does not meet the security criteria, a 400 Bad Request error is returned with details on password requirements.
4. **User Not Found:** If the email provided does not match any user, a 404 Not Found error is returned.

Procedure:

- Click on "Run" to start the automation test. Postman will iterate through each row of your CSV file, substituting the variables in your request with the corresponding values.

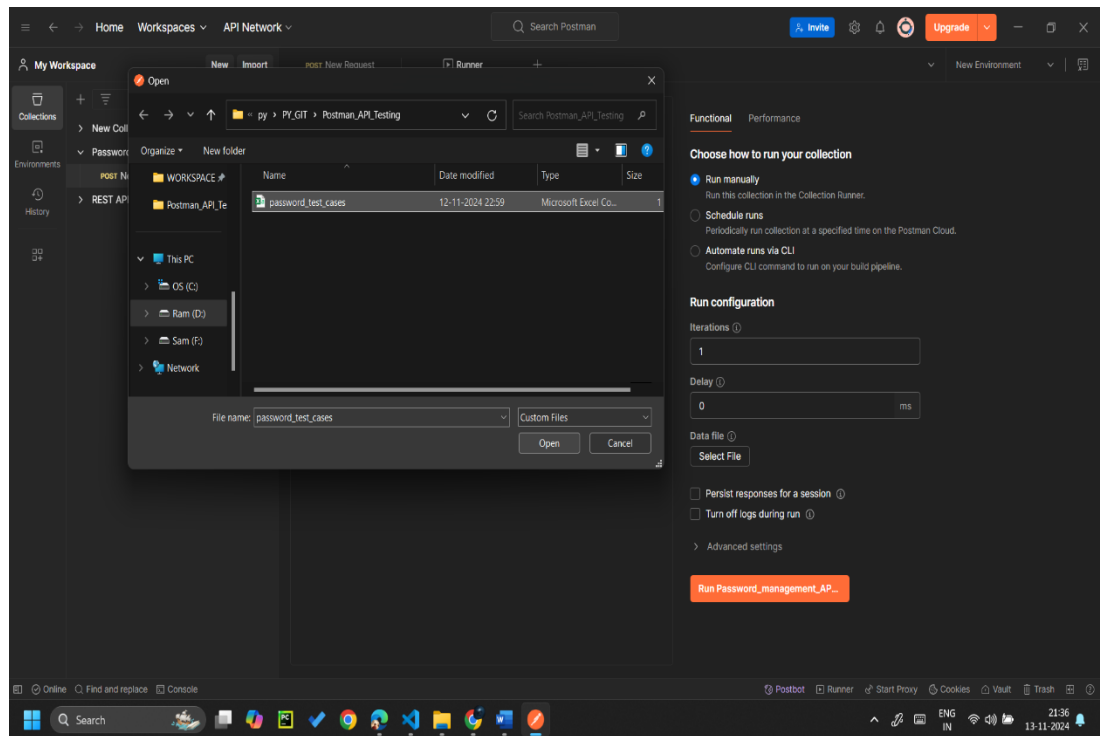


Figure 1.4: Password Test

View Results in Postman:

After running your tests in the Postman Collection Runner, you can review the results to assess the performance of your change-password API with various test cases, including email format validation and handling of empty input fields. Here's how to interpret the results and some content for common scenarios:

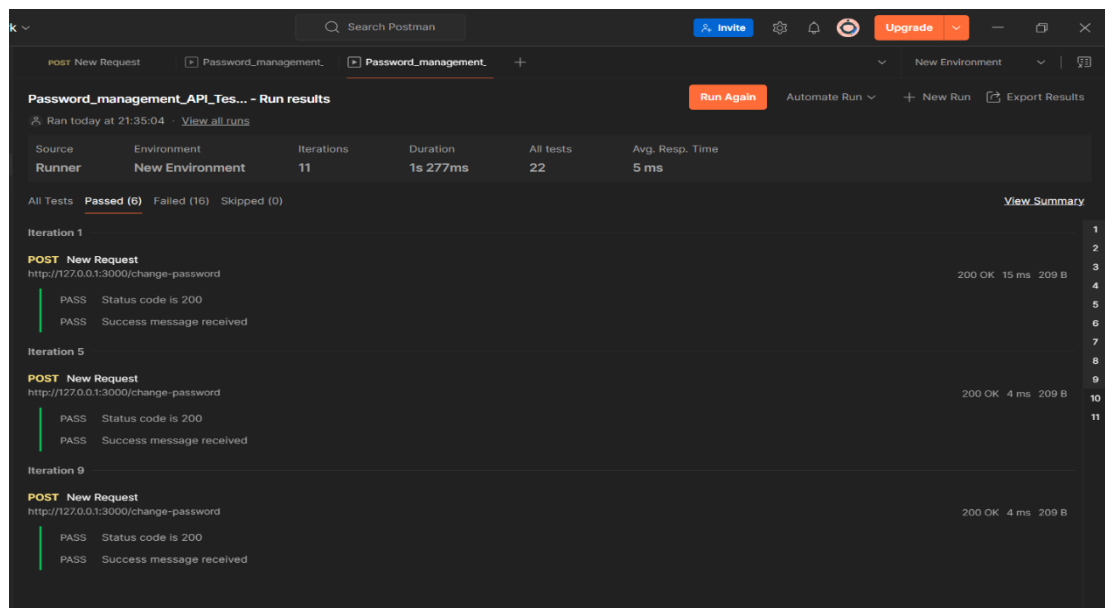


Figure 1.5: Successful Test Cases

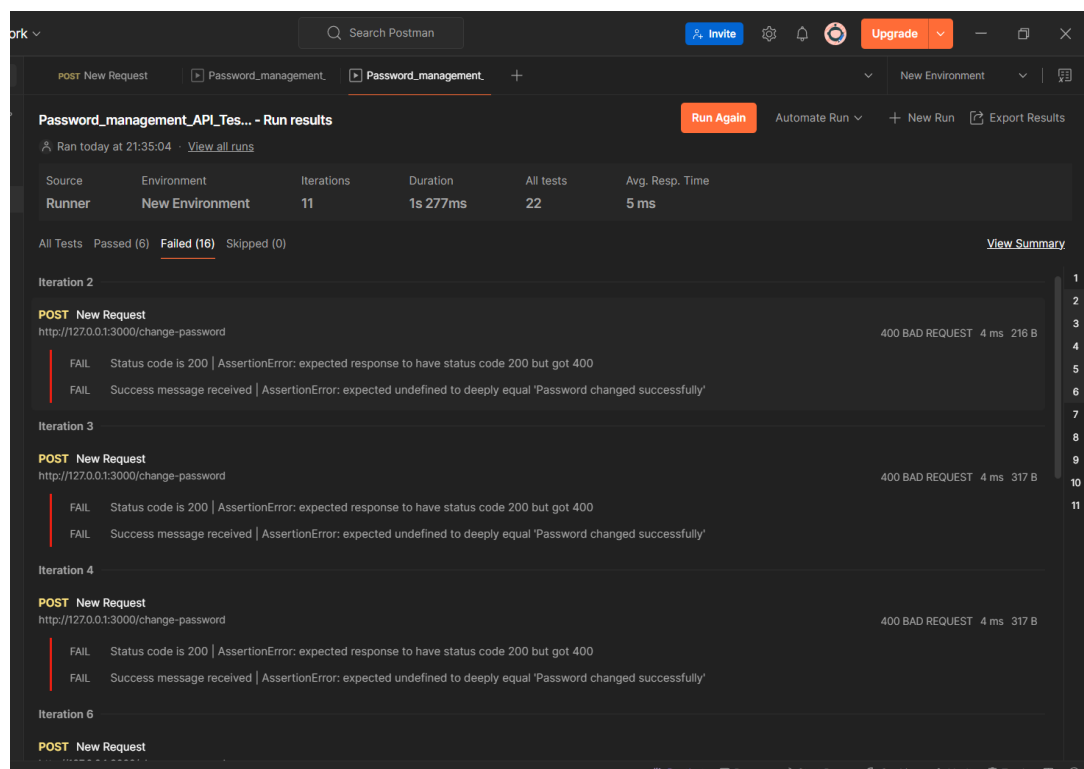


Figure 1.6: Failed Test Cases for the giving input

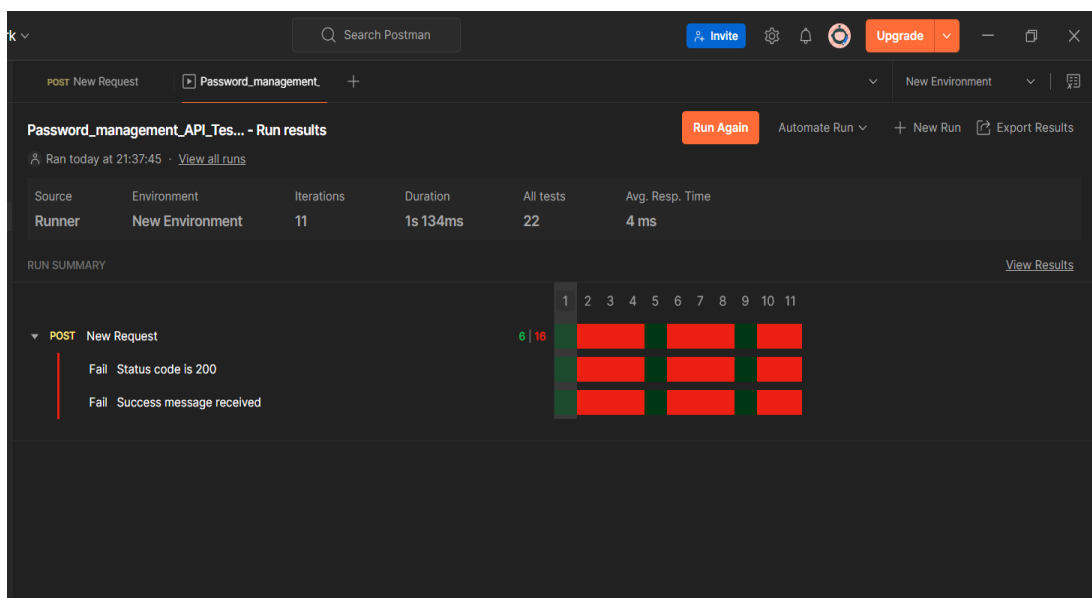


Figure 1.7: All Passes/ Failed Test cases

Conclusion:

By analyzing the results in the Collection Runner, you can ensure your login API behaves correctly under various input scenarios. Testing for valid and invalid email formats, as well as handling empty input fields, are crucial for a robust and user-friendly login experience.

GITHUB LINK:

https://github.com/mathav-ramalingam/Postman_API_Automation_Testing.git