

API Testing Automation Suite (JSONPlaceholder)

Overview

This project demonstrates a complete **API Testing Automation Suite** built using **Postman** and executed via **Newman CLI**.

It validates REST API endpoints of the JSONPlaceholder public API, covering **CRUD operations** for all available resources.

The project showcases:

- Collection-based API test organization
 - Automated test scripts for validation
 - Dynamic data handling across requests
 - CLI execution with detailed HTML reporting
 - End-to-end workflow coverage
-

Tools & Technologies

- **Postman** – API request building & test scripting
 - **Newman** – Command-line collection runner
 - **newman-reporter-htmlextra** – Advanced HTML report generator
 - **JavaScript (Chai assertions)** – Validation logic
-

Test Coverage:

- **Users API** → CRUD operations implemented (GET, POST, PUT, DELETE)
- **Posts API** → CRUD operations implemented
- **Comments API** → CRUD operations implemented
- **Albums API** → CRUD operations implemented
- **Photos API** → CRUD operations implemented
- **Todos API** → CRUD operations implemented

How to Run Locally

1. Clone this repository / download the files

Install dependencies:

```
npm install -g newman newman-reporter-htmlextra
```

2. Run the collection with environment:

```
newman run jsonplaceholder_collection.json -e  
jsonplaceholder_env.json -r cli,htmlextra
```

3. View HTML report in `/newman/` folder.
-

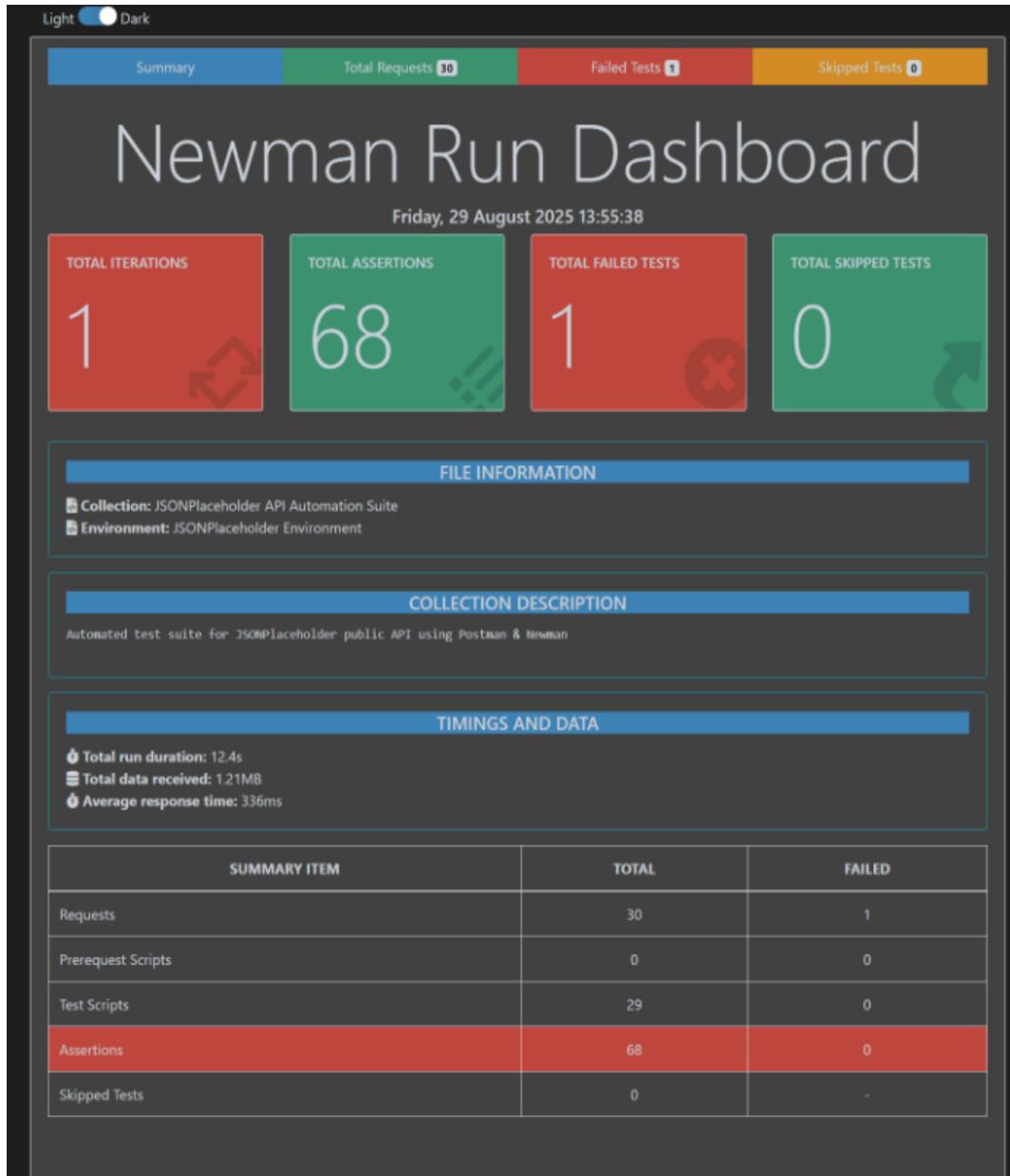
Tests Implemented

- Status code validations (200, 201, 404, etc.)
 - Response body field checks (id, title, body, completed, etc.)
 - Response time checks
 - Negative scenario checks
-

Report:

Below is a snapshot of the Newman Run Dashboard:

- **Total Requests:** 30
- **Total Assertions:** 68
- **Passed:** 67
- **Failed:** 1 (example: response time validation)
- **Skipped:** 0



Key Points

- Organizing API tests into structured collections & folders
- Writing reusable test scripts with Postman
- Running tests headlessly using Newman CLI
- Generating professional HTML reports
- Building a portfolio-ready **end-to-end API Testing project**