

Test Plan – Sprint 1

Project: JSONPlaceholder API Testing

Date: August 9, 2025

Author: Victor Murashev

1. Test Plan Overview

1.1 Purpose

This test plan describes how we will test the API in Sprint 1. The main goal is to check the basic functions of the Posts and Users endpoints.

1.2 Final Goals

- Build automated test framework with Cucumber and TypeScript
- Test all CRUD operations for Posts and Users
- Reach at least 95% test pass rate
- Set up CI/CD pipeline using GitHub Actions
- Make sure tests are stable and repeatable

1.3 Scope

Included:

- Endpoints: /posts, /posts/{id}, /users, /users/{id}
- Test types: Smoke, Functional, Validation
- Test automation
- Basic performance testing (response time)

Not Included:

- Endpoints: Comments, Albums, Photos
- Security testing

- Load testing

2. Test Items

2.1 Features to Test

Feature	Priority	Test Types
Posts CRUD Operations	High	Functional, Validation, Smoke
Users CRUD Operations	High	Functional, Validation, Smoke
API Response Structure	High	Schema Validation
Error Handling	Medium	Negative Testing
Response Times	Medium	Performance

2.2 API Endpoints

Posts:

- GET /posts
- GET /posts/{id}
- POST /posts
- PUT /posts/{id}
- DELETE /posts/{id}

Users:

- GET /users

- GET /users/{id}
- POST /users
- PUT /users/{id}
- DELETE /users/{id}

3. Test Approach

3.1 Strategy

- **API-First Testing:** Focus on backend functionality
- **BDD:** Use Cucumber to describe test scenarios
- **Risk-Based Testing:** Test most important features first
- **Automation-Focused:** Cover at least 90% with automated tests

3.2 Test Types

Type	Coverage
Smoke Tests	10%
Functional Tests	60%
Validation Tests	20%
Performance Tests	10%

3.3 Design Techniques

- Equivalence Partitioning
- Boundary Value Analysis
- Decision Table Testing
- Error Guessing

4. Test Environment

4.1 Environment Details

- **Base URL:** <https://jsonplaceholder.typicode.com>
- **Type:** Production (read-only)
- **Access:** Public, no login needed
- **Data:** Static test data

4.2 Tools and Technologies

- **Framework:** [Cucumber.js](#) + TypeScript
- **HTTP Client:** Axios
- **Test Data:** [Faker.js](#)
- **Reports:** HTML + JSON
- **CI/CD:** GitHub Actions
- **Containerization:** Docker

5. Test Schedule

5.1 Sprint Timeline (12 days)

Week 1:

- Day 1–2: Set up test framework
- Day 3–4: Write tests for Posts API
- Day 5: Write tests for Users API

Week 2:

- Day 6–7: Run and fix tests
- Day 8: Add performance tests
- Day 9: Configure CI/CD
- Day 10: Final test run

Final Days:

- Day 11: Review test reports
- Day 12: Sprint retrospective and planning

5.2 Daily Activities

Day	Activity	Owner	Deliverable
1	Environment setup	Automation Engineer	Running test framework
2	Test data preparation	Test Analyst	Test data sets
3–4	Posts API tests	Automation Engineer	Automated test suite
5	Users API tests	Automation Engineer	Extended test suite
6–7	Test execution	Test Analyst	Test results
8	Performance tests	Automation Engineer	Performance suite
9	CI/CD setup	Automation Engineer	Automated pipeline
10	Final validation	Test Manager	Complete test run

6. Entry, Exit & Suspension Criteria

6.1 Entry Criteria

- [x] Test environment is ready
- [x] Test framework is configured
- [x] Test data is prepared
- [x] Team has access to tools
- [x] API documentation is reviewed

6.2 Exit Criteria

- [x] All planned tests are executed
- [x] At least 95% tests passed
- [x] No critical defects are open
- [x] Response time is under 2 seconds
- [x] CI/CD pipeline works correctly
- [x] Reports are reviewed
- [x] Test artifacts are saved

6.3 Suspension Criteria

- API is down for more than 4 hours
- Test framework has blocking issues
- Major API changes affect test cases

7. Test Deliverables

7.1 Test Documents

- [x] Test Case Specification
- [x] Traceability Matrix
- [x] Environment Setup Guide
- [x] Test Execution Reports

7.2 Test Artifacts

- [x] Cucumber feature files
- [x] Test data generators
- [x] CI/CD config files
- [x] Docker setup

7.3 Reports

- [x] Daily test reports
- [x] Sprint summary report
- [x] Defect reports
- [x] Performance results