

Cypress API Automation Project Guide

Project Concept: "User Management API Testing"

A real-world simulation for testing user registration, login, profile, and role-based access using a free live API.

Live API to Use:

<https://reqres.in> - a great public API for learning and automation.

Test Flow / Feature Scope

Use Case:

Testing a backend API for a User Management System where users can register, login, fetch profiles, update info, and perform role-based operations.

API Endpoints from Reqres:

Action	Method	Endpoint	Notes
Register	POST	/api/register	Requires email, password
Login	POST	/api/login	Returns token
List Users	GET	/api/users?page=2	Fetch users
Get User	GET	/api/users/:id	Get user info
Create User	POST	/api/users	Add user
Update User	PUT	/api/users/:id	Update info
Delete User	DELETE	/api/users/:id	Delete user

Cypress API Automation Project Steps

Step 1: Project Setup

```
npm init -y
npm install cypress --save-dev
npx cypress open
```

Folder Structure:

```
cypress/
- e2e/
- api/
- auth.cy.js
- userCrud.cy.js
- chaining.cy.js
- mock.cy.js
```

Step 2: Test Scenarios

Auth Tests (auth.cy.js)

- Register User using POST /api/register
- Login User using POST /api/login

CRUD Tests (userCrud.cy.js)

- Create, Get, Update, and Delete a user using relevant endpoints.

API Chaining Test (chaining.cy.js)

- Register - Login - Fetch Users with token usage.

Mock API Response (mock.cy.js)

- Intercept GET /api/users?page=2 and mock the response.

Step 3: Reporting

Install: npm install mochawesome --save-dev

Update cypress.config.js with reporter: 'mochawesome'

Run: npx cypress run --e2e

Step 4: CI Integration (GitHub Actions)

Sample .github/workflows/api-tests.yml file to run Cypress tests in CI.

Summary

Area	What You'll Learn
API Testing	Auth, CRUD, headers, assertions
Chaining	Token reuse, multiple requests
Mocking	Intercept & simulate
Reporting	Mocha, HTML reports
CI	GitHub Actions or Jenkins