

Automation Concept: Login Feature

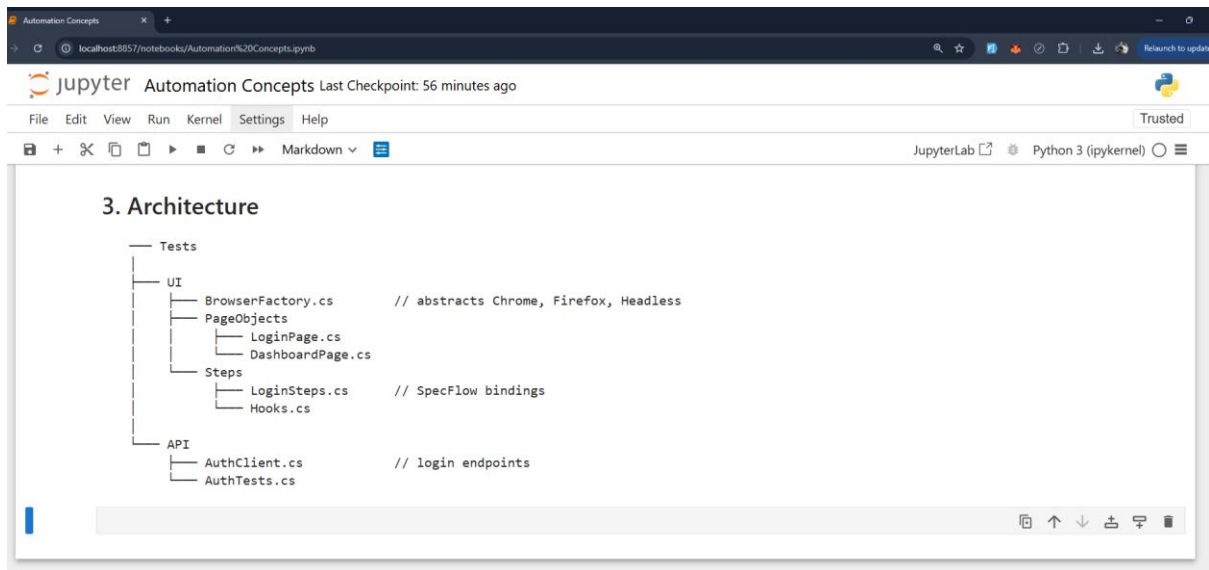
1. Overview

Automate the login flow of our chosen e-commerce site to verify

1. Happy path (valid user)
 2. Edge cases (invalid credentials, locked account, missing fields)
-

2. Tooling

- Primary UI tool:
 - **C# + Selenium WebDriver**
 - Mature bindings
 - Integrates seamlessly with NUnit/xUnit
 - **(Future option: Playwright)** for faster, more reliable cross-browser headless runs
- API tests:
 - **RestSharp** or **HttpClient** in C#
 - Quick feedback loop
 - Validate auth tokens, error codes, rate limits
- UI tests (with BDD layer)
 - **SpecFlow + Gherkin**
 - Readable feature files for stakeholders
 - Hooks into NUnit for parallel execution



4. Data Management

- Test data stored in JSONCSV
- Valid user invalid user blank fields locked account

5. Test Strategy

API Smoke

- **POST** /login - assert 200 OK and validate token schema

API Negative

- Invalid password - assert appropriate error code (e.g. 401 Unauthorized)
- Missing username - assert 400 Bad Request and error message

UI Smoke (fast)

- Direct Selenium/WebDriver calls to verify happy-path login

UI BDD

- SpecFlow Gherkin scenarios covering:
 - Valid credentials
 - Invalid credentials
 - Empty fields
 - Locked/disabled account

6. Reporting & Artifacts

- Screenshots on failure
- HTML Dashboard
- Logging

7. CI/CD Integration

Pipeline (GitHub Actions / Azure DevOps)

1. Restore & Build
2. API Tests - fail fast on errors
3. UI Tests (headless) in parallel
4. Publish
 - Test results
 - Reports & artifacts
5. *(Optional)* Nightly smoke run on main branch