# ENSEK API Test Approach

## Objective

The goal of this testing effort is to verify the core functionality, security, data consistency, and error handling of the ENSEK API. We aim to ensure that all supported operations behave as expected under typical and edge-case scenarios.

## Scope of Testing

In-Scope:

- Functional API testing for:

- /ENSEK/login (authentication)

- /ENSEK/buy (energy purchase via POST and PUT)

- /ENSEK/orders (retrieving order history)

- /ENSEK/reset (resetting test data)

- Data validation (order IDs, timestamps, fuel types)

- Negative testing (invalid energy, negative quantities)

- Pre/post-state checks (e.g., order count before today)

Out-of-Scope:

- Internal database or storage-level validation

- Performance/load testing (unless explicitly extended)

## Testing Types

- Functional: Verifies expected behavior of each endpoint and workflow

- Boundary/Edge: Tests min/max values (e.g., quantity = 0 or -1)

- Negative: Verifies API response to invalid input or unauthorized access

- Data Integrity: Ensures created orders appear correctly

- Authentication: Validates token-based login

- Pre/Post-State: Verifies state changes before/after API calls

## Tools & Frameworks

- .NET 8: Core language

- Reqnroll: BDD framework

- NUnit: Test runner

- RestSharp: HTTP client

- HttpClient: Native HTTP client

- ChromeDriver: UI testing

- Newtonsoft.Json: JSON parsing

## Test Organization

Tags:

- @api – API-only scenarios

- @ui – WebDriver UI tests

- @smoke – Critical-path tests

- @negative – Validation checks

## Structure

/EnsekApiTests

├── Features/

├── StepDefinitions/

├── Hooks/

├── EnsekApiTests.csproj

## Sample Scenarios

Positive Tests:

- Login returns token

- Buying valid fuel returns order ID

- Orders listed in /ENSEK/orders

- Reset clears data

Negative Tests:

- Invalid fuel type

- Quantity = 0 or negative

- Duplicate orders

- Orders after reset = 0

## Test Execution

- Run via: dotnet test

- Filter by tag: --filter TestCategory=api

- CI Integration: GitHub Actions or Azure DevOps

## Acceptance Criteria

- 200 OK on valid calls

- 400 Bad Request on invalid input

- Orders persisted correctly

- 401 Unauthorized without token