

UI Automation Challenge

SUT:

<https://www.themoviedb.org/>

Scenarios:

- **Scenario:** Successful Login
Given the user wants to login
When the user enters the credentials
Then the user should be able to login
- **Scenario:** Failed Login
Given the user wants to log in
When the user submits invalid credentials
Then the user should see a red error message
And the user should see two more error messages
- **Scenario:** Successful Search
Given the user wants to search for a movie
When the user enters the title 'Fight Club'
Then the user should see the movie as the first result
- **Scenario:** Verify Movie Genre Filter
Given the user wants to see the top-rated movies
And the user wants to filter for 'action' movies
When the user applies the 'action' filter
And the user selects any movie
Then the user should see the genre of the movie includes 'action'
- **Scenario:** Validate Acting Timeline
Given the user selects a movie with actors
And the user selects an actor from the top billed cast
When the user checks the actor's acting timeline
Then the title of the movie should be in the timeline

- **Scenario:** Sort by Dates on Ascending Order
Given the user wants to sort top-rated movies by their date
When the user sorts by date on ascending order
Then the user should see the dates of the first 4 movies in ascending order

Examination Criteria

The challenge consists in automating the scenarios provided, applying the concepts learned during the UI-Selenium training. Sharing code with your peers is completely forbidden and will have an entire point deduction in the evaluation of your challenge.

You are allowed to use any test runner, reporting tool and login tool of your choice.

If you want to investigate, you are allowed to use the BDD tool Cucumber as well, however, keep in mind that this is not mandatory for the challenge.

The criteria for the evaluation consist in the following items.

Mandatory:

- Best locator strategies
- Page object model (page object design pattern)
- Meaningful loggers
- Execution of the entire suite using Maven
- Test reporting implementation

Forbidden:

- Thread.sleep()
- Positional locators
- Code duplication
- Hard-coded values

Bonus points:

- Parallel execution of the test cases

- Code compatibility for multiple browser (Look for the factory design pattern and apply it to the driver, this is often referred to as 'Webdriver Factory')
- Screenshots on test failures.
- Execution of the suite in Jenkins

Bonus Test Case: Add Movie to List Using API and UI

1. Create a list using the API
2. Add a movie to that list (use UI or API)
3. Access the list using via UI
4. Verify the movie's title

Any doubts regarding the technical aspect of the challenge, you can ask your mentors, the members of the automation community and me.

If you have any questions regarding the challenge itself don't hesitate to contact me.

Best of Lucks!