

Cypress JavaScript Framework Overview

Automation framework developed using JavaScript +cypress.io(tool) with Page Object Model which can be used across different web-based applications.

We have used “<https://adactin.com/HotelApp/index.php>” url to develop automation tests and these automated test scripts runs with Cypress.io

Framework overview:

- ❖ Adactin
 - pageLocators
 - pageActions
 - testData
- ❖ Cypress
 - fixtures
 - plugins
 - support
 - integration
 - BookHotel.spec.js
 - DateError.spec.js
 - PageValidation.spec.js
 - SearchHotel.spec.js

Details:

pageLocators

- This directory contains all the locators of each action.

pageActions

- This directory contains all the actions that are called in spec file.

testData

- This directory contains all the test data that is used in each action and are called in spec file

Available sample scripts

- pageValidation.spec.js
 - This test case displays the navigation to different pages in the web application.
- dateError.spec.js
 - This test case displays the error message when invalid or past dates are given.
- searchHotel.spec.js
 - This test case searches for required hotel and displays the result.
- bookHotel.spec.js
 - This test case searches for a hotel, books the required hotel and then displays the booking confirmation.

Cypress.io Features

- Time travel
- Debuggability
- Real time reloads
- Automatic waiting
- Spies, stubs, and clocks
- Network traffic control
- Consistent results
- Screenshots and videos

Details:

- Time travel
 - Cypress takes snapshots as your tests run. Simply hover over commands in the Command Log to see exactly what happened at each step.
- Debuggability

- Stop guessing why your tests are failing. Debug directly from familiar tools like Chrome Dev Tools. Our readable errors and stack traces make debugging lightning fast.
- Real time reloads
 - Cypress automatically reloads whenever you make changes to your tests. See commands execute in real time in your app.
- Automatic waiting
 - Never add waits or sleeps to your tests. Cypress automatically waits for commands and assertions before moving on. No more async hell.
- Spies, stubs, and clocks
 - Verify and control the behavior of functions, server responses, or timers. The same functionality you love from unit testing is right at your fingertips.
- Network traffic control
 - Easily control, stub, and test edge cases without involving your server. You can stub network traffic however you like.
- Consistent results
 - Our architecture doesn't use Selenium or WebDriver. Say hello to fast, consistent and reliable tests that are flake-free.
- Screenshots and videos
 - View screenshots taken automatically on failure, or videos of your entire test suite when run headlessly.

Pre-requisites:

Cypress

- Version: 3.6.0 or higher

Java

- Version: 1.8

NodeJs

- Version: 10.16.1 or higher

Steps to use:

Run Tests

- Clone project from repo
 - Git clone
“<https://github.com/mkunduri/zenqautomationsamples.git>”
- You can run the program using any IDE like Visual Studio or WebStorm
- Open terminal from project root path
- Install “package.json” packages using below command
 - npm install
- After installing all packages to open cypress use the below command
 - npm run cypress:open
- After a moment, the Cypress Test Runner will launch.
- Click on required test case to run or click on run all specs to run all tests.

Creating Tests

- Clone project from repo
 - Git clone
“<https://github.com/mkunduri/zenqautomationsamples.git>”
- You can run the program using any IDE like Visual Studio or WebStorm
- Add tests under “cypress/integration” directory