

10th UCAAT

User Conference on
Advanced Automated Testing

Test Automation as a service

Mesut Durukal



Linkedin: mesutdurukal



AGENDA

1

Preparation

Problem Definition

**Requirement
Analysis**

2

Implementation

3

Serve

Maintenance

Support

Chapter One

Preparation

Problem Statement

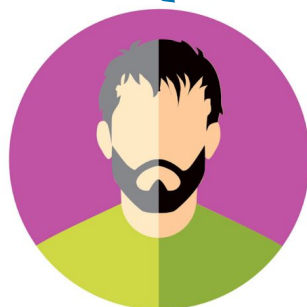
- No standard across multiple teams/projects
- Struggling with problems, which were already solved somewhere else
- Duplication
- Code quality
- Reliability issues
- No monitoring



```
myFramework.checkText(filterButton, '25 k')
```



Nayin



Leo



Susan



Rob

```
cy.get(button + ':visible')
```

```
.click()
```

```
.should('contain', '25 k');
```

```
cy.get(button)
```

```
.should('be.visible')
```

```
.click()
```

```
.should('have.text', '25 k');
```

```
cy.get(button + ':visible')
```

```
.click()
```

```
.invoke('text')
```

```
.as('labelOnButton');
```

```
cy.wait(2000);
```

```
cy.get('@labelOnButton')
```

```
.then((actualLabel) => {
```

```
cy.get(button + ':visible')
```

```
.click()
```

```
.then((button) => {
```

```
  expect(button
```

```
    .text())
```

```
    .to
```

```
    .equal('25k');
```



Targets



- Developing standards.
- Avoid any antipatterns.
- Remove duplication.
- Provide solutions to common problems.
- Provide an opportunity for monitoring
- Dashboards, queries, filters, charts

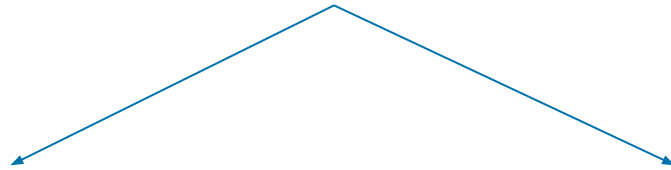


Roadmap

- Objectives & Key Results
- Requirements definition
- Prioritization

Objectives and Key Metrics

QUALITY ASSURANCE



Improve Confidence

Quantity: Coverage

Quality: Escaped Bugs

Support Fast Delivery

Automation

Execution duration

Implementation

False alarms

Test fix time

| | |
|----------------------------------|------------|
| Test coverage | 90% |
| Number of critical incidents | <3 |
| Automation coverage | 100% |
| Total regression duration | <10 mins |
| False failures | <5% |
| Average test fix time | < 24 hours |
| Average test implementation time | < 1 sprint |



Challenges

Safari Automation

No infrastructure for automated regression testing

No proper Jira workflow for bugs

Impossibly testable cases

Unclear requirements / features

Infrastructure instabilities

Errors raised by the SUT

Cross domain redirection

Code changes to selectors / missing selectors

Proctor tests roll ups breaking automation

Test flakiness

Escaped bugs

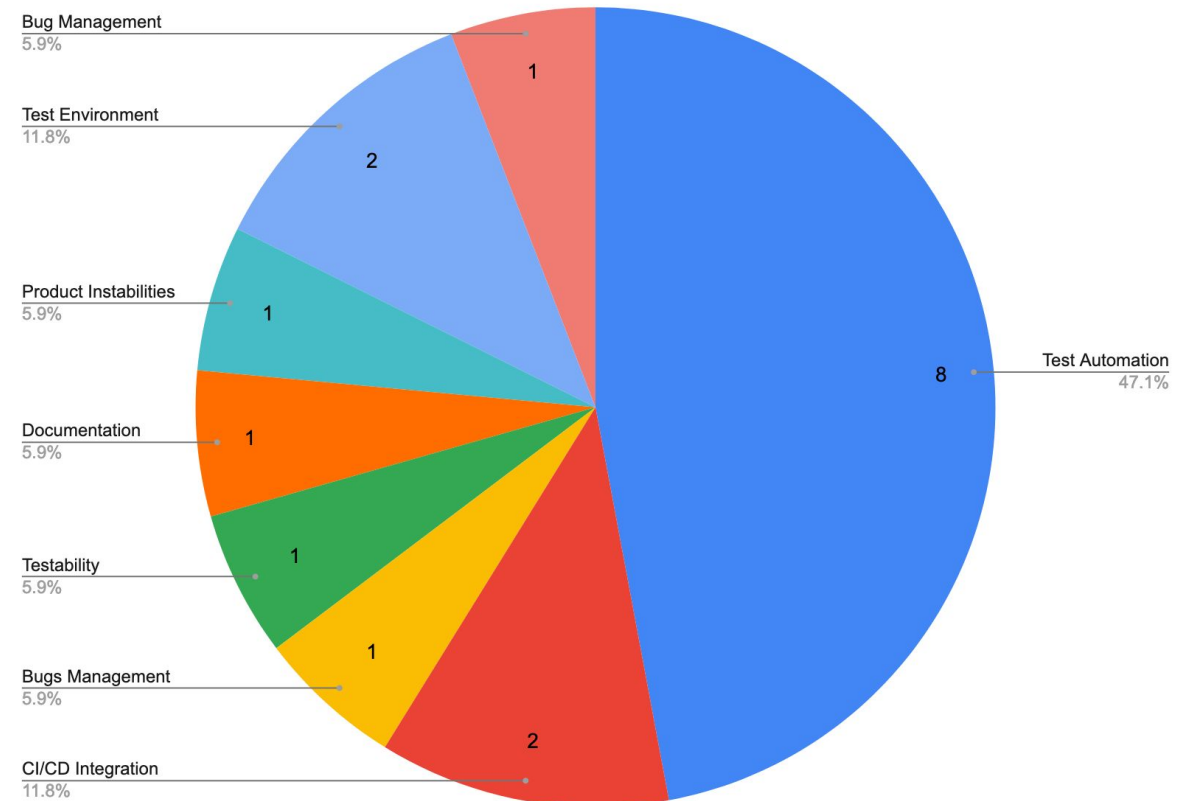
Share tests between similar products

Use same approach for tests within a project

Keeping account profile data untainted

API testing framework compatibility

Categories of Pitfalls



Requirements

Testing BE and/or FE together should be supported

Rest API and GraphQL testing should be supported

API testing should support authenticated requests

FE testing modules should support Cookie operations

Including individual or a group of tests to the suite

Excluding tests from the executions should be possible.

Requests modification/interception should be supported

Response modification/interception should be supported

Test Input Management should be in place

Automatic bug reports after failures

Slack Notifications after executions

Evidence collection

Monitoring dashboards

There should be Hard fail and soft fail modes

Tests should be easily integrable to Gitlab/Github

Tests should be executable on multi-branch pipelines

Tests should be executable on different branches

Tests should be executable after commits/before merges

Cross domain testing should be supported

Clean up

Mocking should be supported

Various browsers should be supported

Mobile testing should be supported (native app)

User manual should be generated

It should be easy to write new tests

Parallel execution should be supported

Test flakiness should be detected and reduced

Retry (line/whole test) strategy

No sensitive data should be revealed

Resources should be configurable

Static code analysis (linters, servers)

Dependencies should be auto-updated?

Accessibility testing components

Tests should have priorities

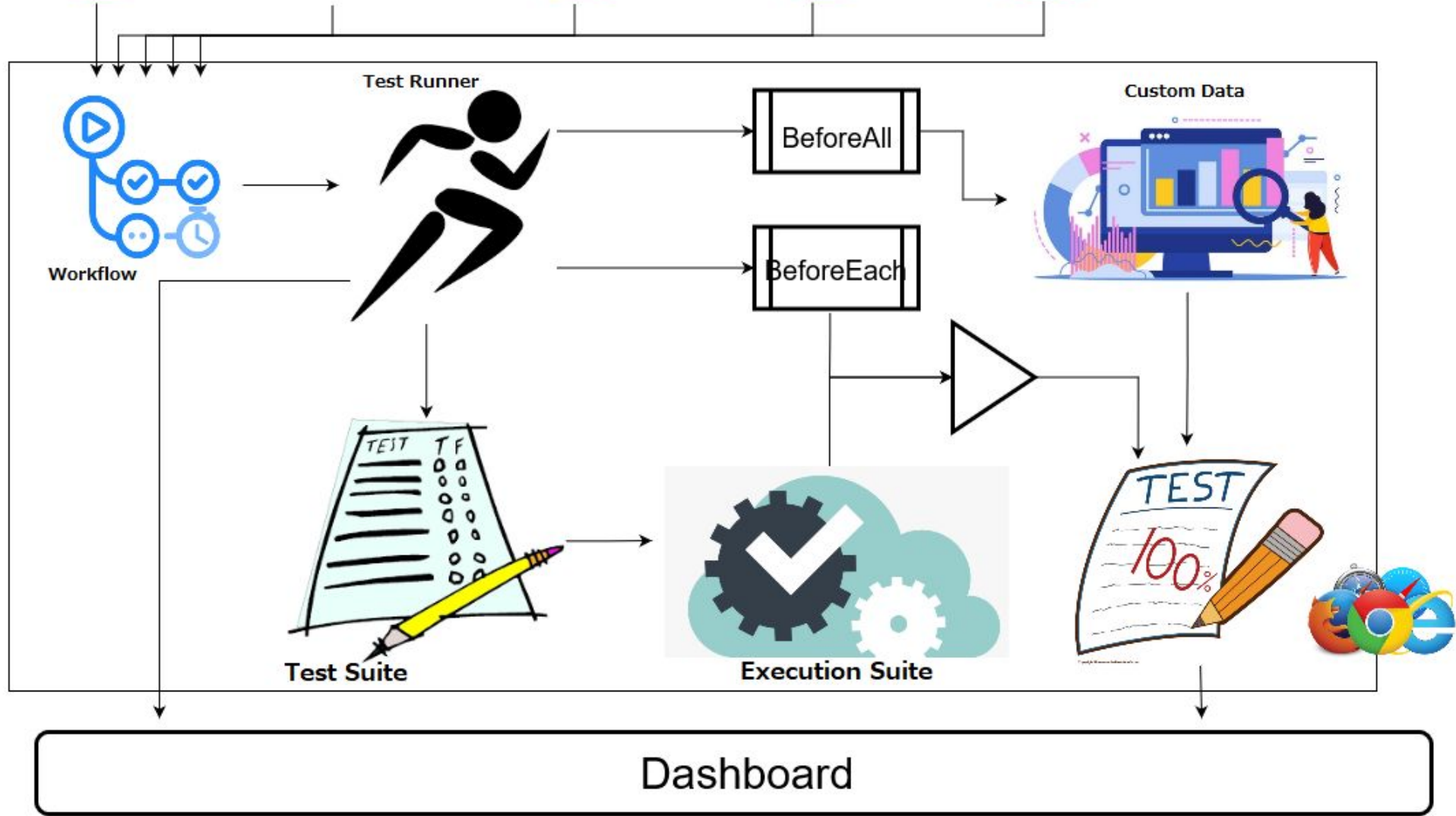
Async requests should be handled properly.

Chapter Two

Implementation



Multiple Applications under Test



Benchmarking

- Community Research
- Check other benchmarks
- Self experience
- Decision Criteria
 - Speed
 - Ease of coding
 - Flexibility
 - Documentation & Support
 - Licensing / Cost
 - Feature Compatibility



Selenium

```
describe('Search Button Test', () => {
  it('Search button test', async () => {
    console.log('Initialize');
    let driver = sbase.driver;
    let query = seleniumLocators.keywordInput();
    let keywordBox = await sbase.getElementById(query);
    let searchButton = await sbase.getClassName(
      seleniumLocators.searchButton()
    );

    console.log('Search Button is disabled before filling queries');
    expect(await searchButton.getAttribute('disabled')).toEqual('true');

    console.log('Click Search button after entering a query');
    let randomQuery;
    do {
      randomQuery = base.generateRandomText();
    } while ('0x'.includes(randomQuery.toLowerCase()));
    await keywordBox.sendKeys(randomQuery);
    await searchButton.click();

    console.log('Check that button navigates to results');
    keywordBox = await sbase.getElementById(query);
    expect(await keywordBox.getAttribute('value')).toEqual(randomQuery);
    expect(await driver.getCurrentUrl()).toContain('q=' + randomQuery);
  }, 15000);
});
```

Nightwatch

Testcafe

```
fixture('Search Button Test')
  .page(userVariables.url)
  .requestHooks(CustomHeader);

test
  .meta('testid', 'C31881136')
  .meta('platform', 'executionPlatform')
  .meta('priority', 'p0')('Search Button Test', async

  console.log('Initialize: ' + mytest.fixtureCtx.myVa
  const select = new selectors();
  await mytest.resizeWindow(320, 568).setNativeDialog

  console.log('Search Button is disabled before filli
  await mytest
    .expect(select.find(select.searchButton()).hasA
    .ok());

  console.log('Click Search button after entering a q
  let randomQuery;
  do {
    randomQuery = new TestBase(
  } while ('0x'.includes(randomQu
  await mytest
    .typeText(select.keyword(),
    .click(select.searchButton(

  console.log('Check that button
  await mytest
    .expect(select.find(select.
    .eql(randomQuery)
    .expect(await select.curren
    .contains('q=' + randomQuer

  describe('Search Butt
  this.tags = [
    {
      testid: 'C
      platform:
      priority:
    },
  ], 15000);
  it('Search button test', function (browser) {
    //console.log('Initialize');
    const select = browser.page.selectors();

    //console.log('Search Button is disabled before filling queries');
    select.assert.attributeEquals('@searchButton', 'disabled', 'true');

    //console.log('Click Search button after entering a query');
    let randomQuery;
    do {
      randomQuery = base.generateRandomText();
    } while ('0x'.includes(randomQuery.toLowerCase()));
    select.setValue('@keyword', randomQuery).click('@searchButton');

    //console.log('Check that button navigates to results');
    select.assert.attributeEquals('@keyword', 'value', randomQuery);
    browser.assert.urlContains('q=' + randomQuery);
  });
});
```

Cypress

```
describe('Search button', function () {
  Cypress.env().executionPlatforms.forEach((executionPlatform) => {
    it(
      'Search button ' + executionPlatform,
      {
        testid: 'C31881136',
        platform: executionPlatform,
        priority: 'p0',
      },
      () => {
        base.log('Initialize');
        let performSearch = new PerformSearch();

        base.log('Search Button is disabled before filling queries');
        cy.get(serpPage.searchButton()).should('be.disabled');

        base.log('Click Search button after entering a query');
        let randomQuery;
        do {
          randomQuery = base.generateRandomText();
        } while ('0x'.includes(randomQuery.toLowerCase()));
        performSearch.test('Search Button Test', async ({ page }) => {
          cy.get(serpPage
            test.info().annotations.push({
              testid: 'C31881136',
              platform: ['desktop', 'mobile'],
              priority: 'p0',
            });
          console.log('Initialize');
          let keyword = page.locator(select.keywordInput(

          console.log('Search Button is disabled before f.
          await page.locator(select.searchButton()).isDis

          console.log('Click Search button after entering a query');
          let randomQuery;
          do {
            randomQuery = new TestBase().generateRandomText();
          } while ('0x'.includes(randomQuery.toLowerCase()));
          await keyword.type(randomQuery);
          await page.locator(select.searchButton()).click();

          console.log('Check that button navigates to results');
          await expect(keyword).toHaveAttribute('value', randomQuery);
          await expect(page.url()).toContain('q=' + randomQuery);
        });
      });
    });
  });
});
```

Playwright

Quality Dimensions

Understandable

Transparent

Documentation

Troubleshooting



Performance

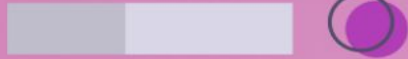
Resource
consumption

Multiple execution

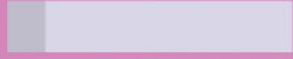
Versioning



Flexible



Integrity



Scalable



Priority

Reliable

Coverage

Non Disruptive

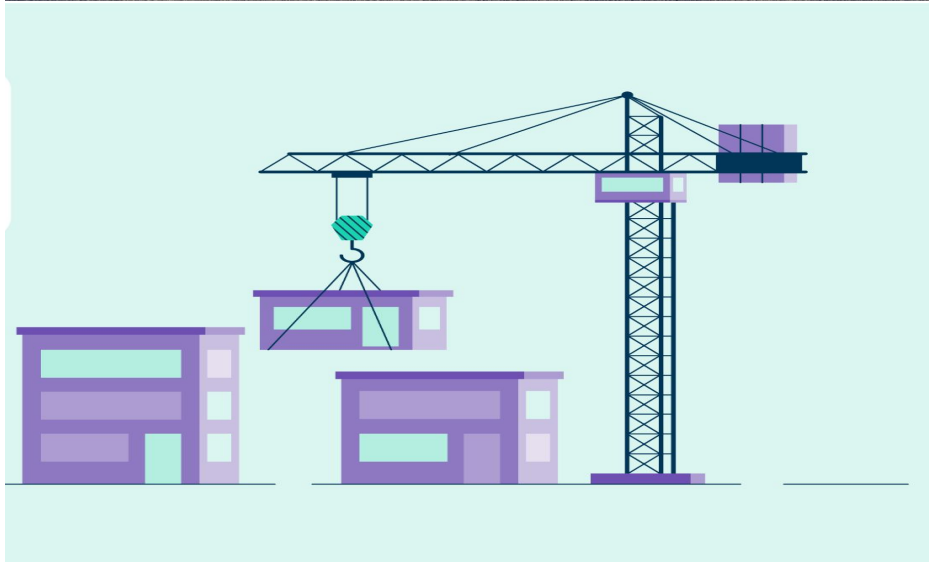
Testability



Flexible: Configurable, Cross Platforms



Integrity: Can integrate to CI platforms



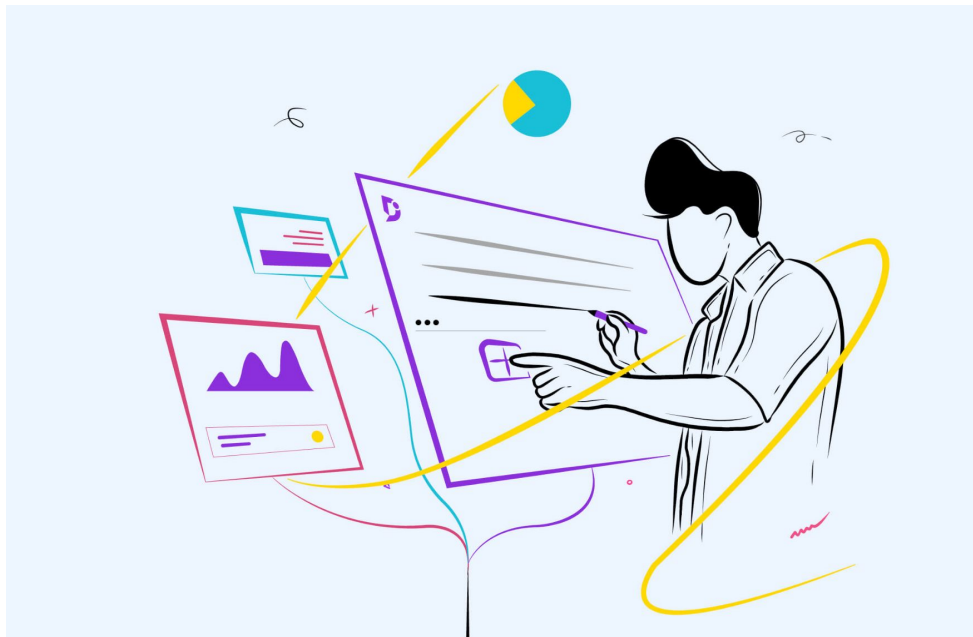
Scalable/Portable: Parallelization, Desktop/Mobile

Goal Oriented: Priorities

Understandable



Transparent, Visible



Documentation: Read Me, Manuals

Troubleshooting, Evidences: SS, Video

Reliable



Coverage



Nondisruptive



Testability: Mocks

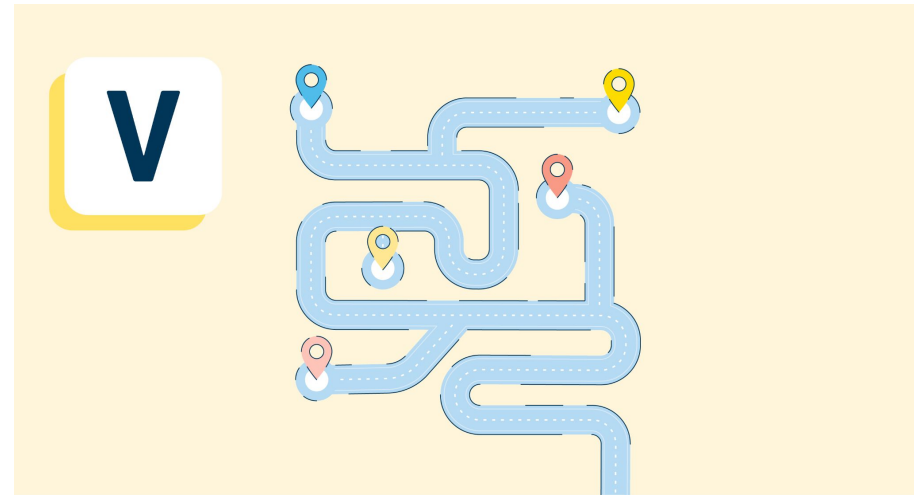
Performance



Resource Consumption



Multiple Execution



Versioning

Chapter Three

Service

Speed Up Implementation

- Define the locators
 - Find the parent of an element
 - Find siblings and children
- Implement test steps
 - Wait for responses, conditions
 - Parse Promises

```
page.locator(selector).click();  
vs  
page.click(page.locator(selector));
```

```
page.locator(':text("label")');  
vs  
page.getElementsByName("label");
```

```
recentSearchesTitle().locator("xpath=..");  
vs  
page.locator("article:has(" +  
recentSearchesTitle() + ")")
```

```
▼ <div class="css-xede2x"> flex  
  <p class="chakra-text css-1gvl1h7">Your Recent Searches</p>  
  ▼ <div class="css-l9j7tk"> flex  
    ▶ <a class="css-1xpn9ty" href="https://www.simplyhired.com/search?q=34&l="> ... </a> event flex  
    ▶ <a class="css-1xpn9ty" href="https://www.simplyhired.com/search?q=&l=56"> ... </a> event flex  
    ▼ <a class="css-1xpn9ty" href="https://www.simplyhired.com/search?q=dsfdsfdsfdsf&l=dsfdsfdsfdsfdsf"> event flex  
      <span class="css-1mrbynz">dsfdsfdsfdsf</span> overflow  
      <span class="css-fxcujc">in dsfdsfdsfdsfdsf</span> overflow  
      <span class="css-5blk2w"></span> overflow  
      <span class="css-17xejub"></span> overflow  
      ▶ <span class="css-aujr02"> ... </span>  
    </a>
```


Eventually, looks like:

```
import { expect, test } from "@playwright/test";

test.describe("My Second Suite", () => {
  test( title: "Search Button Test", testFunction: async ({ page : Page }) => {
    console.log("Initialize");
    let keyword = "myQuery";
    let location = "myLocation";
    await page.click( selector: '[data-testid="mobileFindJobsKeywordButtonToggle"]');
    await page.goto( url: "https://simplyhired.com/");

    console.log("Enter a query");
    await page.locator( selector: '[data-testid="findJobsKeywordInput"]').type(keyword);
    await page.locator( selector: '[data-testid="findJobsLocationInput"]').type(location);

    console.log("Click Search button");
    await Promise.all( values: [
      page.waitForResponse( urlOrPredicate: (resp : Response) =>
        resp.url().includes("search.json?q=myQuery&l=")
      ),
      page.click( selector: '[data-testid="findJobsSearchSubmit"]'),
    ]);

    console.log("Navigate to results");
    await expect(
      page.locator( selector: '[data-testid="findJobsKeywordInput"]')
    ).toHaveAttribute( name: "value", keyword);
    await expect(
      page.locator( selector: '[data-testid="findJobsLocationInput"]')
    ).toHaveAttribute( name: "value", location);
    await page.waitForURL( url: "**/search?q=" + keyword + "**");
  });
});
```

```
1 import { test } from "@playwright/test";
2 const SearchJobs = require("../SearchJobs");
3
4 test.describe("My Second Suite", () => {
5   test( title: "Search Button Test2", testFunction: async ({ page : Page }) => {
6     await page.goto( url: "https://simplyhired.com/");
7     await new SearchJobs(page, app: "simplyhired", platform: "mobile").queryJobs(
8       keyword: "myQuery",
9       location: "locations"
10    );
11   });
12 });
13
```

Measure the effect!

Define Locators

Edit Add comment Assign More Close Issue Reopen Issue Verify

Details

Type: Sub-task Status: **PENDING VERIFICA...** (View Workflow)
Priority: Minor Resolution: Fixed
Labels: **test_automation**
Sprint: FW Polishing (SP4: 8-27/mar)

Description

First implement raw test case

Attachments

Drop files to attach, or [browse](#).

Activity

All Comments Work Log History Activity Transitions

| Transition | Time In Source Status | Execution Times |
|---|-----------------------|-----------------|
| Mesut Durukal made transition - 1 week ago PENDING TRIAGE → ACCEPTED | 2m 22s | 1 |
| Mesut Durukal made transition - 6 days ago IN PROGRESS → ACCEPTED | 2h 16m | 4 |

Average:

2 hours locators

3 hours
implementation



30 mins

Monitoring

Most Occurring Exceptions

The container is too small to display the entire cloud. Tags might be cropped or omitted.

selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash from unknown error: cannot determine loading status from tab crashed (Session info: headless chrome=83.0.4103.61)

selenium.common.exceptions.NoSuchElementException: Message: Could not locate element with visible text: Rejected

TypeError: expected string or bytes-like object

selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash from tab crashed (Session info: headless chrome=83.0.4103.61)

Failure Types Pie

Exception Types

| failure_type.keyword: Descending | exception_message.keyword: Descending | Count |
|----------------------------------|--|-------|
| Unknown | selenium.common.exceptions.InvalidSessionIdException: Message: invalid session id | 42 |
| Unknown | TypeError: expected string or bytes-like object | 4 |
| Unknown | selenium.common.exceptions.NoSuchElementException: Message: Could not locate element with visible text: Rejected | 3 |
| WebDriver Exception | selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash from unknown error: cannot determine loading status from tab crashed (Session info: headless chrome=83.0.4103.61) | 2 |
| Type Error | TypeError: expected string or bytes-like object | 2 |
| Selenium Invalid Session | selenium.common.exceptions.InvalidSessionIdException: Message: invalid session id | 2 |
| Unknown | selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash from tab crashed (Session info: headless chrome=83.0.4103.61) | 2 |
| Unknown | TypeError: 'NoneType' object is not subscriptable | 2 |
| Session Deleted - Page Crashed | selenium.common.exceptions.WebDriverException: Message: unknown error: session deleted because of page crash from tab crashed (Session info: headless chrome=83.0.4103.61) | 1 |

Automation Search

| test_name: Descending | test_answered_questions[en_CA] | Build | Jenkins Job | filters | running sauce | runtime (ms) | Count |
|-----------------------|--------------------------------|-------|--|---------------|---------------|--------------|-------|
| test_name: Descending | test_answered_questions[en_CA] | 66 | https://jenkins.qa.indeed.net/smb-intl/job/smb-intl--automation-schedulers/job/smb-intl--automation-manual-runs/ | status:FAILED | false | 19,304.867 | 1 |

Indeed Employer
Manage candidates from your phone
FREE

Candidates

Aaron Adams

CONTACTED

Message Call Interview More

APPLICATION INTERVIEWS NOTES

Reply quickly with a template

Still interested? Commute

TRAINING



Wrap Up

- Observe Problems, Improvement Areas
- Requirements
- Architecture
- Implementation
- Outcome
- Monitoring



Any further questions?



mesutdurukal



mesutdurukal



DurukalMesut



mesutdurukal.com

