

# ***How to Write Reusable Tests***

---



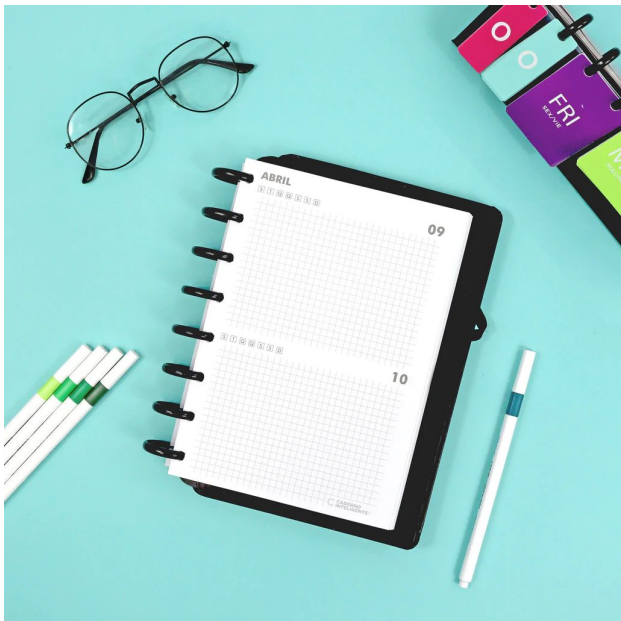
mesutdurukal



DurukalMesut



mesutdurukal.com



01

## Introduction

Duplication vs Reusability

02

## Implementation

Config  
Hook  
Helper Classes  
Locator Strategy  
Test Config  
Interceptions  
Loop

03

## Demo

# Chapter One

---

## INTRODUCTION

# Challenges in Testing



Assertion Roulette

Conditional Test Logic

Constructor Initialization

Exception Handling

Duplicate Assert

General Fixture

Default Test

Eager Test

Empty Test

Ignored Test

Frequent updates in Agile

Test Smells



Magic Number Test

Mystery Guest

Redundant Print

Redundant Assertion

Resource Optimism

Sensitive Equality

Sleepy Test

Unknown Test


Lazy Test


# Duplication: Making More Tricky



# Reusability



 area, station

 job title, keyword, company

search

Salary


Employment Form


Occupation

Job characteristics



keep list

 area, station

 job title, keyword, company

search

Salary


Employment Form


Occupation

Job characteristics



keep list

 area, station

 job title, keyword, company

search

Salary

Employment Form

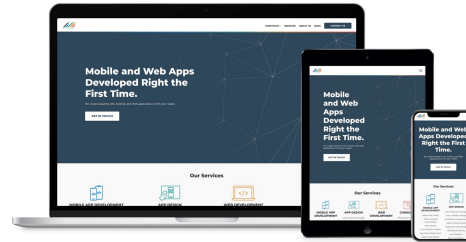
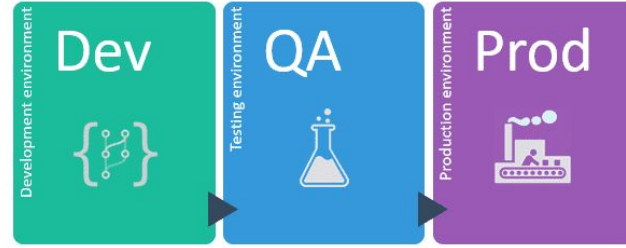
Occupation

Job characteristics



keep list

# Compatibility



## Chapter Two

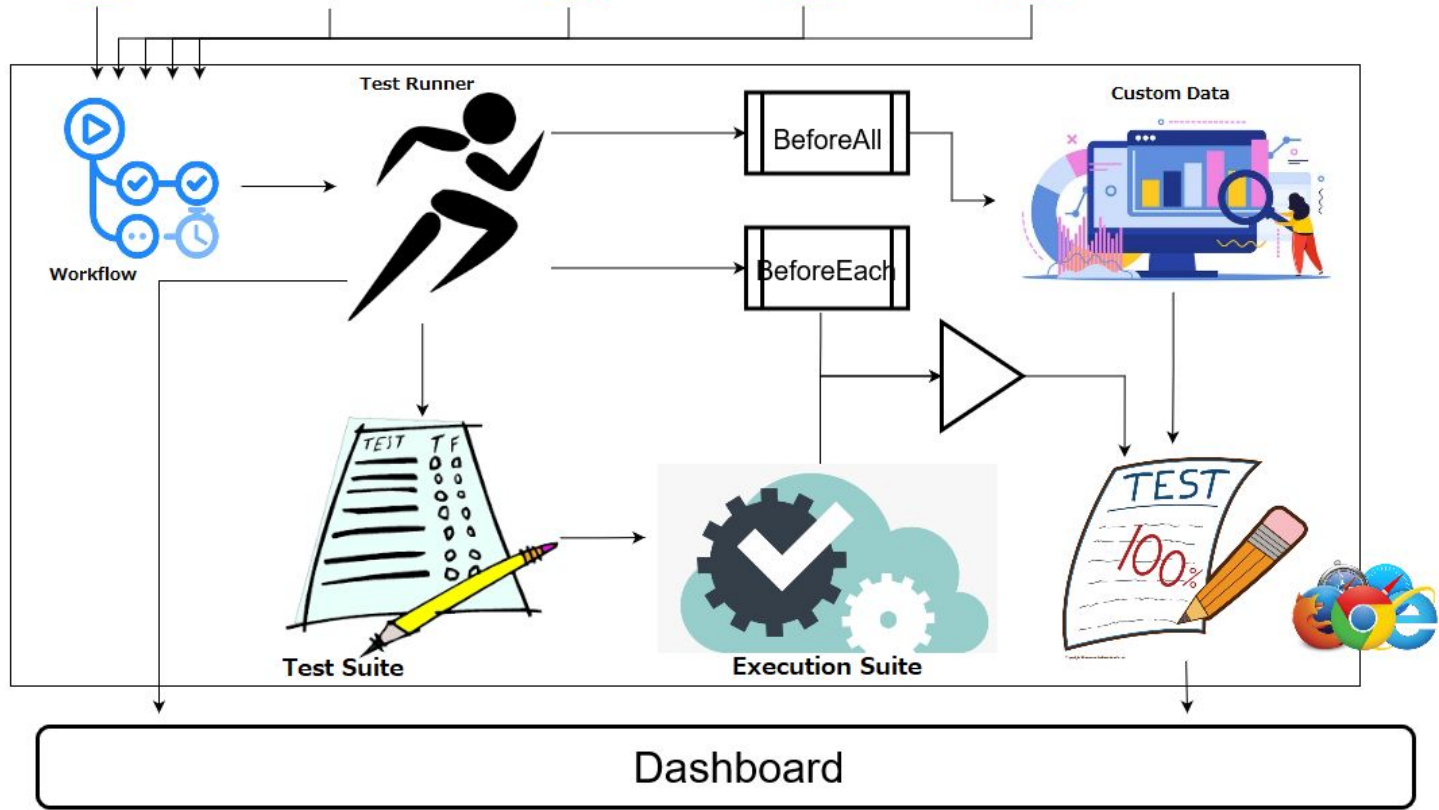
---

### Quick Reminder for Key Features





Multiple  
Applications  
under Test



2 lines  
Running  
in 48  
configurations

```
describe( title: 'Query Search Autocomplete', fn: function () {  
  it( 'Query Search Autocomplete ' + executionPlatform,  
    () => {  
    base.log('Perform a random search to see autocompletion');  
    let randomQuery = base.generateRandomText();  
    cy.get(serpPage.keywordInput()).clear().type(randomQuery);  
  
    base.log('Auto complete options should show up');  
    cy.get(serpPage.query_autocomplete());  
  }  
});
```

Terminal: Local x + v

```
mdurukal@IT-JPN-14954 % node_modules/.bin/cypress run --spec "QuerySearchAutocomplete.spec.js"
```

chrome,firefox,edge

--browser edge

--env app=townwork

twm,rhk,rnn,fan

qa,prod

stage=prod

executionPlatforms=["desktop","mobile"]<sup>10</sup>

```
{
  "defaultCommandTimeout": 5000,
  "env": {
    "myKey": "myValue",
    "myApp": "car"
  },
  "fixturesFolder": "./cypress/fixtures",
  "integrationFolder": "./cypress/demo-folder",
  "pluginsFile": "./cypress/plugins",
  "retries": {
    "runMode": 1,
    "openMode": 0
  },
  "projectId": "ta7vk2",
  "supportFile": "./cypress/support",
  "screenshotsFolder": "./cypress/screenshots",
  "video": false,
  "videosFolder": "./cypress/videos",
  "redirectionLimit": 50,
  "experimentalSessionAndOrigin": true,
  "chromeWebSecurity": false
}
```

# Excluding Config from Test Code

cypress open

cypress run

--config-file ./my\_custom\_config.json

--env some\_Variable=myVariable

--browser chrome --headed

--spec "./cypress/myWorkingFolder/\*\*"

Cypress.env()

# Hooks

## Support > index.js

```
before(fn: function () {});  
  
beforeEach(fn: function () {});  
  
after(fn: function () {});  
  
afterEach(fn: function () {});
```

Order: First global hooks, then in-class hooks

# Classes, Functions

```
class MyClass {  
    myVar = 'something';  
    my_function() {}  
}  
export default MyClass;
```

```
import importName from './myClass';  
new importName().my_function();
```

```
globalThis.commonClass = new a();
```

# Page Interactions

```
cy.visit()
```

navigate

```
cy.url()
```

read current URL

## Locate elements

```
cy.get()
```

```
.children()
```

```
.contains()
```

```
.find()
```

```
.eq()
```

```
.each()
```

## Interactions

```
.click()
```

```
.clear()
```

```
.type()
```

```
.check()
```

```
.uncheck()
```

```
.select()
```

## Attributes

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

```
.text()
```

```
.length
```

## Assertions

```
.should('contain', 'expText')
```

```
.should('have.attr', 'href')
```

```
.should('have.class', 'expClass')
```

```
.should('have.value', 'expValue')
```

```
.should('have.length', 2)
```

```
.should('be.empty')
```

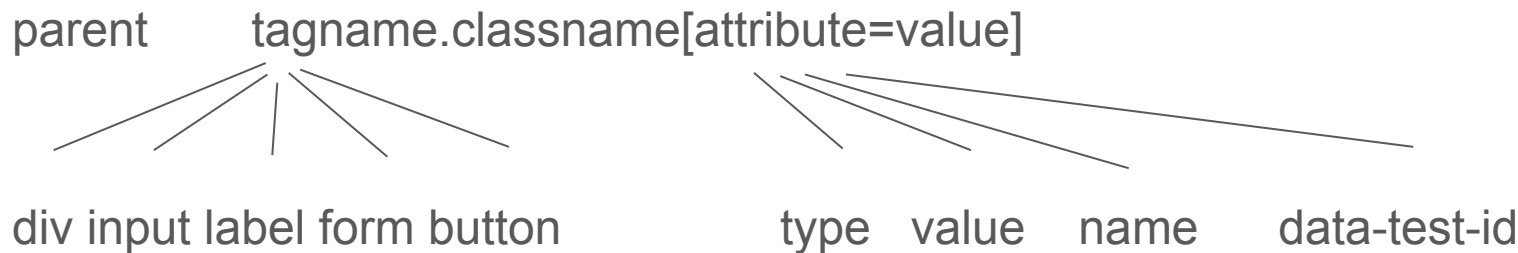
```
.should('be.visible')
```

```
.should('be.checked')
```

```
.should('be.greaterThan', 3)
```

```
.should('include', 'expectedText');
```

# Locators & Selectors



We can narrow the selection criteria with element attributes:

- `:contains("text")`
- `:visible`
- `:nth-child(N)`

# Chapter Three

---

## Demo



# A very simple scenario

Visit WebApp landing page

Type in the query text field

-> User query should be accepted by the system

Click Search button

# Start Implementation!

```
cy.visit('https://www.careesma.in/');  
cy.get('div input[name="q"]').type('mytext');  
cy.get('div input[name="q"]').should('have.value', 'mytext');  
cy.get('[data-testid = "findJobsSearchSubmit"]').click();
```

# Locators

```
cy.visit('https://www.careesma.in/');  
cy.get('div input[name="q"]').type('mytext');  
cy.get('div input[name="q"]').should('have.value', 'mytext');  
cy.get('[data-testid = "findJobsSearchSubmit"]').click();
```

duplication

```
let keyword field = locators.textField  
let search button = locators.searchButton
```

```
cy.visit(main page);  
cy.get(keyword field).type('mytext');  
cy.get(keyword field).should('have.value', 'mytext');  
cy.get(search button).click();
```

# Run on multiple apps

```
globalThis.mainPage =  
  Cypress.env().app +  
  '-' +  
  Cypress.env().stage +  
  '.cortexsdk.com/';
```

```
"env": {  
  "app": "careesma"  
},
```

```
--env app='wor'
```

# Test Config

```
it( title: 'Test Name', config: {}, fn: () => {
```



```
it('Test Name', { priority: 'p1' }, () => {})
```

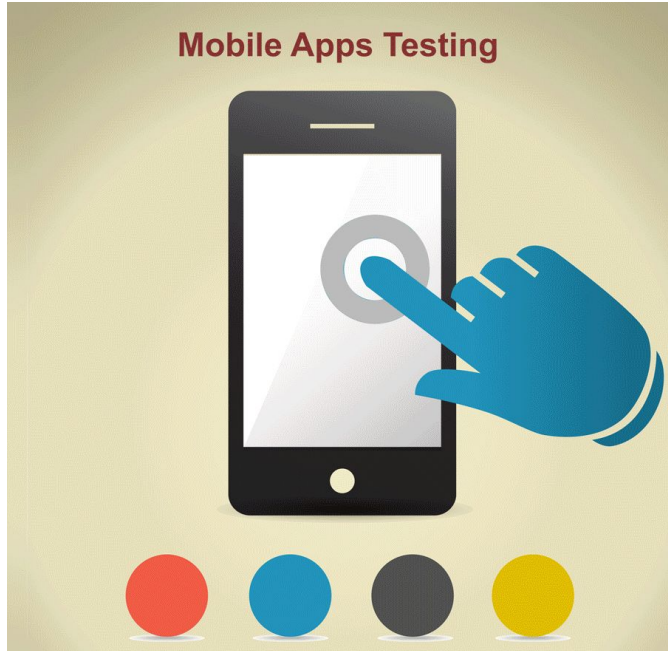
```
this.currentTest.testConfig.unverifiedTestConfig.priority
```

```
this.skip();
```

# Test Config

```
it(  
    'Job type Autofill All Keywords ' + executionPlatform,  
    {  
        test_id: 'C30970008',  
        platform: executionPlatform,  
        priority: 'p3',  
        app: 'townWork,fromA',  
    },  
)
```

# Mobile Testing



```
if (platform === 'mobile') {  
  cy.viewport('iphone-5');  
}
```

# Intercepts

```
cy.intercept('https://www.careesma.in/**',  
  (req) => {req.headers['user-agent'] = this.mobileAgent;}  
).as('addHeader');
```

```
cy.intercept('*/api/keepList?*) .as('AddKeepList');
```

## ▼ TEST BODY

1	visit	https://www.careesma.in/	
	(document)	● GET 200 /	req modified addHeader
2	get	#query-input	
3	- type	mytext	
▶	(ping)	● POST 200 ...	req modified addHeader 2
	(fetch)	● GET 200 /a...	req modified addHeader
	(fetch)	● GET 200 /a...	req modified addHeader
	⚙️ (fetch)	● GET /api/a...	req modified addHeader
	(fetch)	● GET 200 /a...	req modified addHeader



# Loop

```
for (let i = 0; i < count; i++) {}
```

```
['desktop', 'mobile'].forEach((executionPlatform) => {  
  Cypress.env().executionPlatforms.forEach((executionPlatform)=> {  
    it('Test Name', { platform: executionPlatform }, () => {  
      });  
    });  
  });  
});
```

