Automation testing tool selection and implementation

Tool we are integrating for the testing is **cypress**

1. Introduction to Cypress

- Cypress is the next generation automation testing tool
- Open source
- Built on node js
- javaScript is used to write the test cases

2. Advantages of Cypress

- Easy to install and use
- Fast growing community
- Deliver fast, reliable and consistent execution
- Provide debuggability
- Asynchronization
- Custom commands can be added to remove redundancy

3. Type of Test that Cypress support

- End to end test
- Integration test
- Unit test
- Anything that runs inside the browser

4. Browser in Cypress

By default it uses the browser called **electron** which is a chromium version for electron. You can pass the browser of your own choice by using the following command

- npx cypress run --browser=chrome --headless

List of browsers that cypress support

- Chrome 64 and above.
- Edge 79 and above.
- Firefox 86 and above.

5. Install of Cypress

5.1. Prerequisite

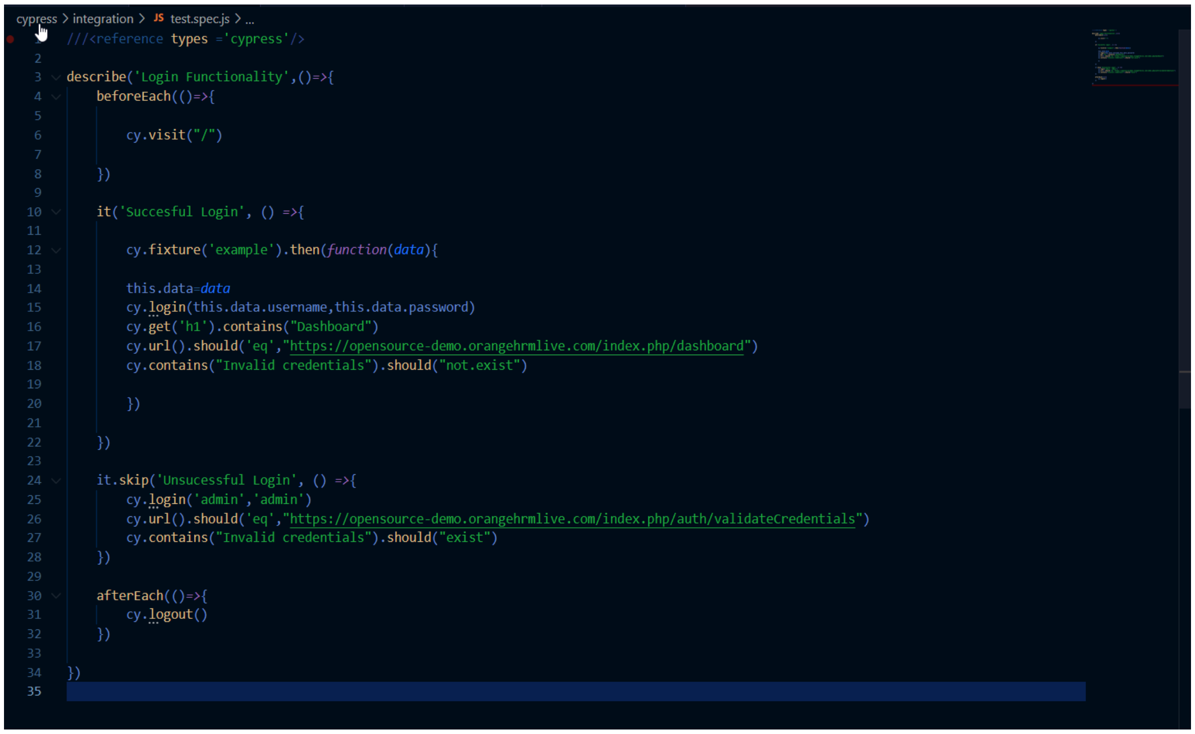
- node js version should be install
- node js should be passed in the environment variable
- Visual studio code should be install

5.2. Installation of Cypress

- open the folder in the VS code
- Run the following command
    - npm install cypress
- once the installation is finished, run the following command to run the cypress locally
    - npx cypress open
- Remove all the existing test case example form the following folder
    - cypress-->integration
- Add the new file test.spec.js under the following folder
    - cypress-->integration

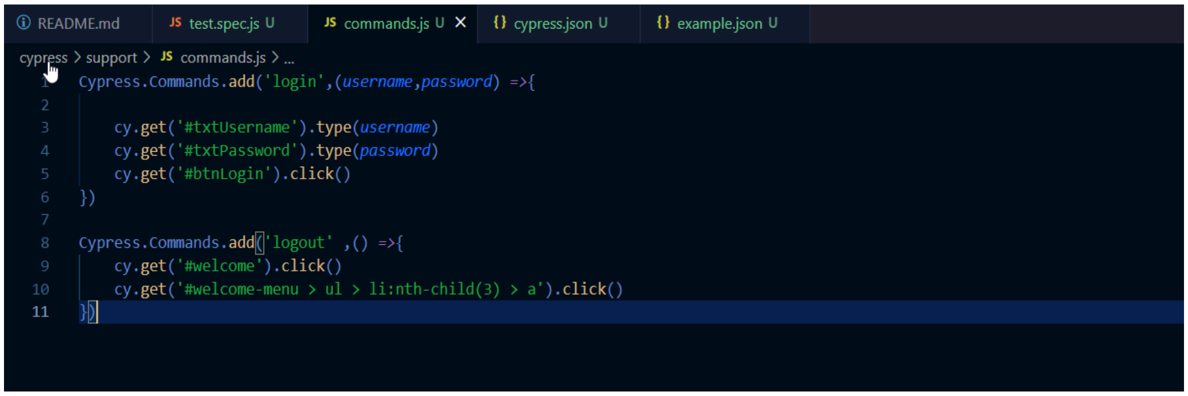
6. Writing Test cases in the test.spec.js

- Adding the dummy test script in the code



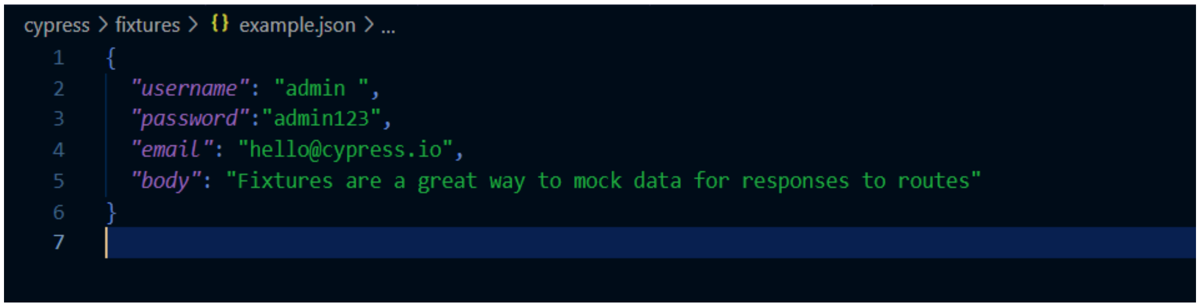
```
///<reference types ='cypress'/>

describe('Login Functionality',()=>{
    beforeEach(()=>{

        cy.visit("/")

    })

    it('Succesful Login', () =>{

        cy.fixture('example').then(function(data){

        this.data=data
        cy.login(this.data.username,this.data.password)
        cy.get('h1').contains("Dashboard")
        cy.url().should('eq',"https://opensource-demo.orangehrmlive.com/index.php/dashboard")
        cy.contains("Invalid credentials").should("not.exist")

        })

    })

    it.skip('Unsucessful Login', () =>{
        cy.login('admin','admin')
        cy.url().should('eq',"https://opensource-demo.orangehrmlive.com/index.php/auth/validateCredentials")
        cy.contains("Invalid credentials").should("exist")
    })

    afterEach(()=>{
        cy.logout()
    })

})

```

- Custom commands in cypress are used to reduce the redundancy in the code. Adding the command in command.js that are required to be common in test script



```
Cypress.Commands.add('login',(username,password) =>{

    cy.get('#txtUsername').type(username)
    cy.get('#txtPassword').type(password)
    cy.get('#btnLogin').click()
})

Cypress.Commands.add('logout' ,() =>{
    cy.get('#welcome').click()
    cy.get('#welcome-menu > ul > li:nth-child(3) > a').click()
})
```

- Add common data in json file and by using fixture in test script call that data

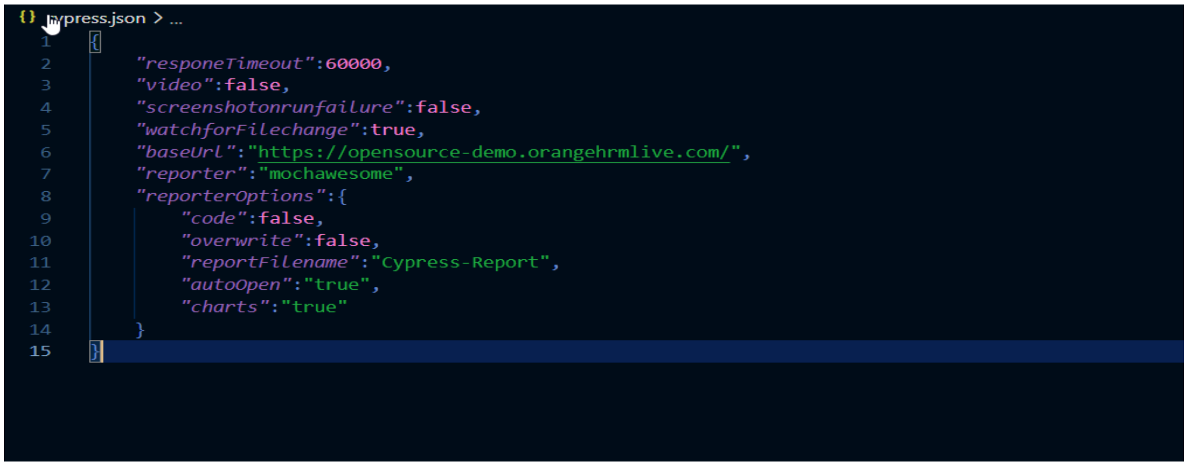


```
{
  "username": "admin ",
  "password":"admin123",
  "email": "hello@cypress.io",
  "body": "Fixtures are a great way to mock data for responses to routes"
}
```

- Adding **hooks** for example **beforeEach** and **afterEach**. Hooks are added to specify what actions need to perform before and after of each test case execution

7. Adding the configuration file

- Add the configuration in the cypress.josn



```
{
    "responeTimeout":60000,
    "video":false,
    "screenshotonrunfailure":false,
    "watchforFilechange":true,
    "baseUrl":"https://opensource-demo.orangehrmlive.com/",
    "reporter":"mochawesome",
    "reporterOptions":{
        "code":false,
        "overwrite":false,
        "reportFilename":"Cypress-Report",
        "autoOpen":"true",
        "charts":"true"
    }
}
```

8. Executing the cypress from command line

Following command is used to execute the cypress test cases from command line

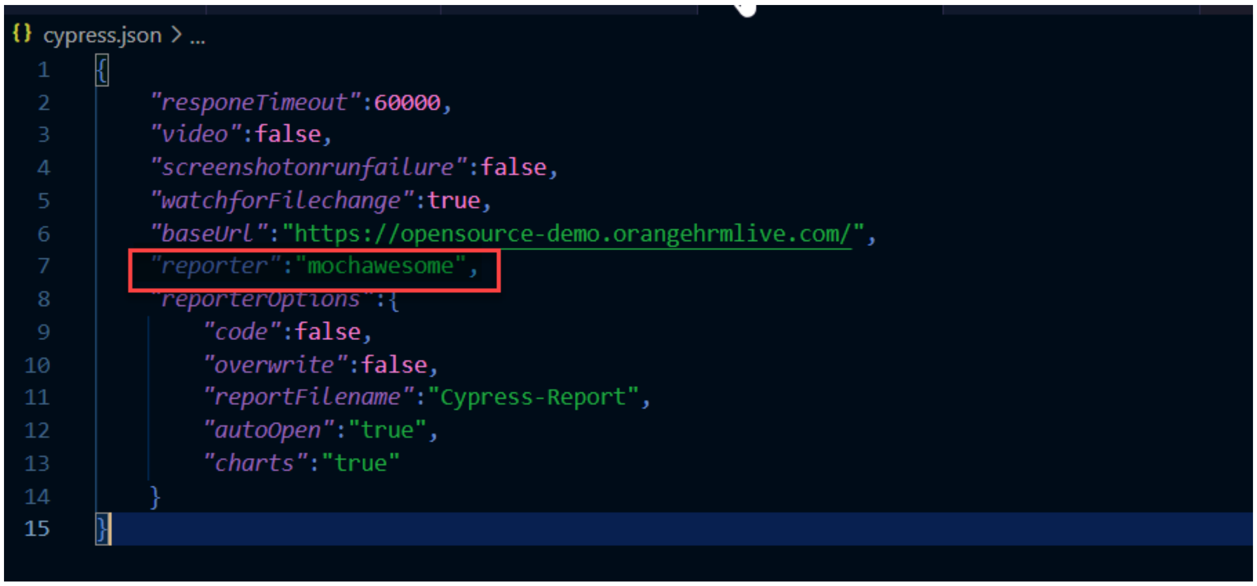
- npx cypress run

9. Integrating Mocha report with

Execute the following command on terminal

- npm install mochawesome

Add the reference to Cypress.json



```
{
    "responeTimeout":60000,
    "video":false,
    "screenshotonrunfailure":false,
    "watchforFilechange":true,
    "baseUrl":"https://opensource-demo.orangehrmlive.com/",
    "reporter":"mochawesome",
    "reporterOptions":{
        "code":false,
        "overwrite":false,
        "reportFilename":"Cypress-Report",
        "autoOpen":"true",
        "charts":"true"
    }
}
```