

mukesh cy-6

first install mocha awesome reporter.

```
npm i --save-dev cypress-mochawesome-reporter
```

Edit config file (`cypress.config.js` by default)

```
reporter: 'cypress-mochawesome-reporter',  
require('cypress-mochawesome-reporter/plugin')(on);
```

Add to `cypress/support/e2e.js` import
`'cypress-mochawesome-reporter/register';`

to run cypress via cmd -

```
npx cypress run
```

to customise the report more like this -

```
1 reporterOptions: {  
2     charts: true,  
3     reportPageTitle: 'custom-title',  
4     embeddedScreenshots: true,  
5     inlineAssets: true,  
6     saveAllAttempts: false,  
7 },
```

codesnap.dev

add them in cypress config.js under reporter.

```
1 const { defineConfig } = require("cypress");
2
3 module.exports = defineConfig({
4
5   reporter: 'cypress-mochawesome-reporter',
6   reporterOptions: {
7     charts: true,
8     reportPageTitle: 'custom-title',
9     embeddedScreenshots: true,
10    inlineAssets: true,
11    saveAllAttempts: false,
12    autoOpen:true,
13    overwrite:false,
14    code:false
15  },
16  e2e: {
17    setupNodeEvents(on, config) {
18      // implement node event listeners here
19      require('cypress-mochawesome-reporter/plugin')(on);
20    },
21    watchForFileChanges:true,
22    retries:{
23      runMode:2,
24      openMode:3
25    }
26  },
27 });
28
```

```
1 // ****
2 // This example support/e2e.js is processed and
3 // loaded automatically before your test files.
4 //
5 // This is a great place to put global configuration and
6 // behavior that modifies Cypress.
7 //
8 // You can change the location of this file or turn off
9 // automatically serving support files with the
10 // 'supportFile' configuration option.
11 //
12 // You can read more here:
13 // https://on.cypress.io/configuration
14 // ****
15
16 // Import commands.js using ES2015 syntax:
17 import './commands'
18
19 require('cypress-xpath')
20 import "cypress-real-events/support";
21 import 'cypress-file-upload';
22 import 'cypress-mochawesome-reporter/register';
```

codesnap.dev

this and arrow function cannot be used together. this and anonymous or normal functions work.

```
1  {
2    "name": "Using fixtures to represent data",
3    "email": "hello@cypress.io",
4    "body": "Fixtures are a great way to mock data for responses to routes"
5  }
6
```

codesnap.dev

```
1  // 
2 describe('fixture for json file', () => {
3
4   beforeEach(function(){
5
6     cy.fixture("example.json").then(function(jsondata){
7       this.jsondata=jsondata;
8     })
9
10  });
11
12
13
14 //when we use arrow function we get this error -
15 //Cannot read properties of undefined (reading 'jsondata')
16   it('fixture first case', function() {
17     cy.log(this.jsondata.name)
18   });
19});
```

codesnap.dev

```
1  {
2      "name": "johny",
3      "email": "johnrambo@test.com",
4      "gender": "female",
5      "coursedescription": "new testing course",
6      "mobile": 7896542310,
7      "subjects": "karan atreya"
8 }
```

```
1  //
2 describe('create delete courses', function () {
3
4     beforeEach(function(){
5         cy.fixture("createuser").then(function(jsondata){
6             this.jsondata=jsondata;
7         })
8
9     });
10
11    //go to the website and try to fill few fields using fixture.
12
13    it('create course', function()  {
14
15        cy.visit
16        ("https://www.tutorialspoint.com/selenium
17 /practice/selenium_automation_practice.php");
18            cy.get("#name").type(this.jsondata.name)
19 .should("have.value", this.jsondata.name);
20            cy.get("#email").type(this.jsondata.email)
21 .should("have.value", this.jsondata.email);
22            cy.xpath("//div[3]//div[1]//div[1]//div[2]//input[1]").click();
23            cy.get("#mobile").type(this.jsondata.mobile)
24 .should("have.value", this.jsondata.mobile)
25            cy.get("#subjects").type(this.jsondata.subjects)
26 .should("have.value", this.jsondata.subjects)
27
28 });
29 });
```

codesnap.dev

to write utility functions use support and command.js files-

```
1 //this is fully hard coded.
2 Cypress.Commands.add("handleautosuggestionlists", function(){
3     //get all the option values
4         //iterate values and if you get value click on it.
5         //we dont have text directly in cypress.
6         //we use jquery.
7         //use each. we can pass all three arguments or any one.
8         //same like for each. jquery, index, list.
9         //wrap - converts any damn element or object
10    //or jquery element to cypress element.
11        cy.xpath("//div[@role='option']/div[1]//span")
12        .each(function(ele){
13            cy.log(ele.text()); //text comes from jquery.
14
15            if(ele.text().includes("api")){
16                cy.wrap(ele).click({force:true}) //convert
17 //to cypress and then click.
18            }
19        })
20    })
21
```

```
1  //  
2 describe('auto suggest google search', () => {  
3  
4     it('select value from google search', () => {  
5  
6         cy.visit("https://www.google.com/")  
7  
8 //         cy.wait(10000);  
9  
10 //           // Try clicking the "Stay signed out" or close button  
11 // cy.contains('Stay signed out').click({ force: true })  
12 // .should('not.exist');  
13  
14 cy.wait(10000);  
15  
16 cy.xpath("//textarea[@id='APjFqb']").click()  
17  
18     //type like robot.  
19     cy.xpath("//textarea[@id='APjFqb']")  
20 .type("mukesh otwani", {delay:200})  
21  
22     //calling the utility command.  
23     cy.handleautosuggestionlists()  
24  
25  
26  
27  
28     });  
29 });
```

```
1 //lets make above code generic without hard coding.
2 Cypress.Commands.add("handleautosuggestionlists",
3   function(locator, textosearch){
4     //get all the option values
5     //iterate values and if you get value click on it.
6     //we dont have text directly in cypress.
7     //we use jquery.
8     //use each. we can pass all three arguments or any one.
9     //same like for each. jquery, index, list.
10    //wrap - converts any damn element or object or
11    //jquery element to cypress element.
12    cy.xpath(locator)
13      .each(function(ele){
14        cy.log(ele.text()); //text comes from jquery.
15
16        if(ele.text().includes(textosearch)){
17          cy.wrap(ele).click({force:true}) //convert to
18        //cypress and then click.
19        }
20      })
21  })
```

```
1  //  
2  describe('auto suggest google search', () => {  
3  
4      it('select value from google search', () => {  
5  
6          cy.visit("https://www.google.com/")  
7  
8          // cy.wait(10000);  
9  
10         // Try clicking the "Stay signed out" or close button  
11         // cy.contains('Stay signed out').click({ force: true }).should('not.exist');  
12  
13         cy.wait(10000);  
14  
15         cy.xpath("//textarea[@id='APjFqb']").click()  
16  
17         //type like robot.  
18         cy.xpath("//textarea[@id='APjFqb']")  
19         .type("mukesh otwani", {delay:200})  
20  
21         //calling the utility command but this time not hard coded.  
22         cy.handleautosuggestionlists  
23         ("//div[@role='option']/div[1]//span", "api")  
24  
25  
26  
27  
28     });  
29 });
```

```
1 //lets try to do for login
2 Cypress.Commands.add("login", function(username, password){
3     cy.get("#username").type(username)
4     cy.get("#password").type(password)
5     cy.get("#submit").should("be.enabled").click()
6 })
```

codesnap.dev

```
1 //
2 describe('login test', () => {
3
4     //use the utility function.
5     it('test valid login', () => {
6         cy.visit("https://practicetestautomation.com/practice-test-login/")
7
8         cy.login("student","Password123");
9
10    });
11});
```

codesnap.dev