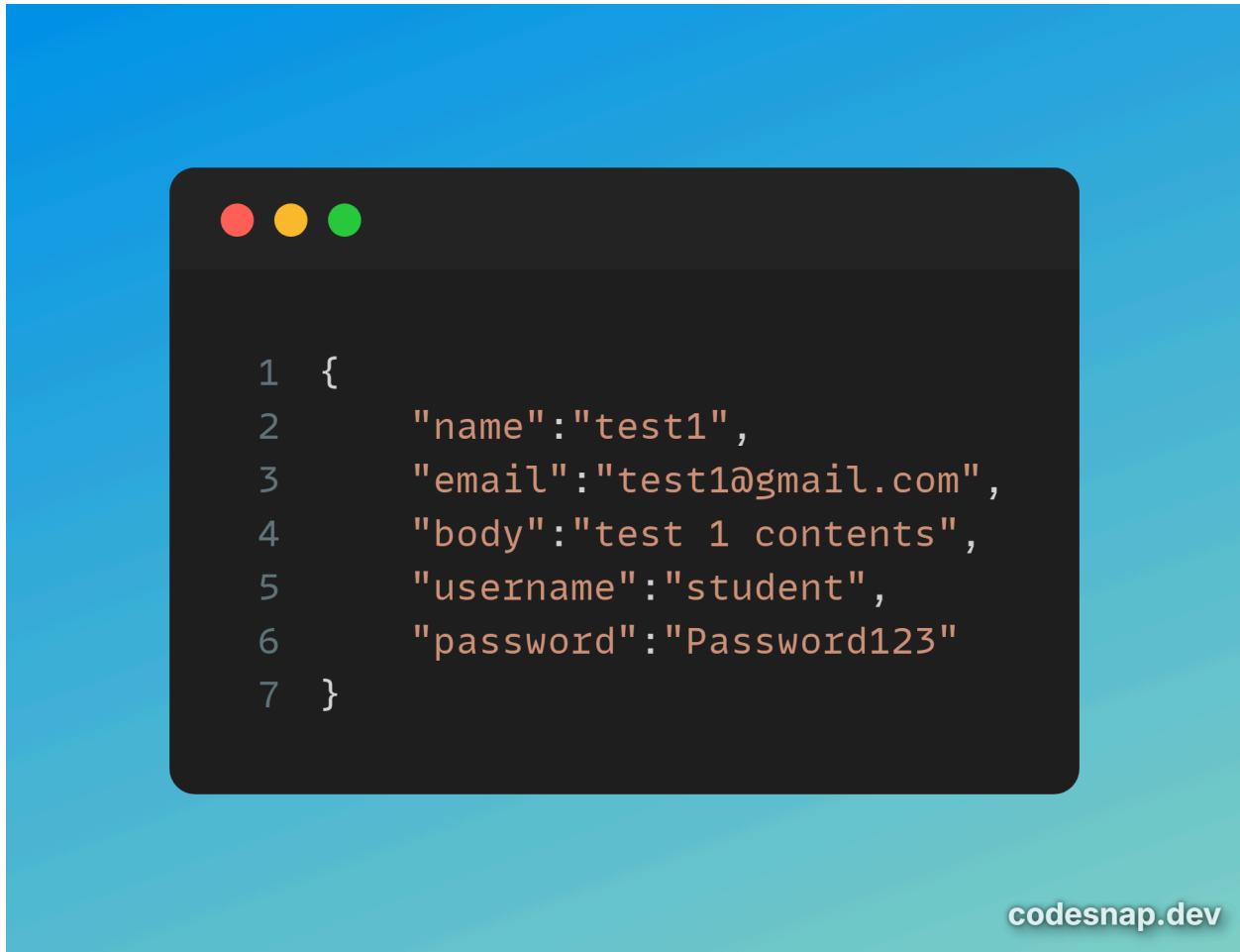


# mukesh cy-7

multiple test data files



A screenshot of a mobile application interface. At the top, there are three colored dots (red, yellow, green) in a horizontal bar. Below this is a large black rectangular area containing white text. The text is a JSON object with seven lines of code, each preceded by a line number (1 through 7). The JSON object contains fields for name, email, body, username, and password.

```
1  {
2      "name": "test1",
3      "email": "test1@gmail.com",
4      "body": "test 1 contents",
5      "username": "student",
6      "password": "Password123"
7 }
```

codesnap.dev



```
1  {
2      "name": "test2",
3      "email": "test2@gmail.com",
4      "body": "test 2 contents",
5      "username": "student",
6      "password": "Password123"
7 }
```

codesnap.dev

```
1 //use multiple fixtures for testing
2 //declare array
3 //context for us to know which test is running like tc number.
4 //we need outer describe block for the entire suite.
5 //inner describe block for the iterations.
6 const testvalueinarray=[
7
8     {
9         "name": "test1",
10        "context" :"1"
11    },
12    {
13        "name": "test2",
14        "context" :"2"
15    }
16 ]
17
18
19
20 describe('ddt for cypress', function(){
21     testvalueinarray.forEach(function(fixturedata){
22         //get the context using fixture data as it holds the complete data.
23         describe(fixturedata.context, function(){
24             // lettestdata;
25             //use before to load fixtures like we used before each.
26             //get the test data one by one.
27             before(function(){
28                 cy.fixture(fixturedata.name).then(function(testdata) {
29                     this.testdata=testdata;
30                 })
31             })
32
33             it("login to app", function(){
34                 cy.log(thistestdata.name);
35                 cy.log(thistestdata.email);
36                 cy.log(thistestdata.body);
37
38                 //lets try to login
39                 cy.visit("https://practicetestautomation.com/practice-test-login/");
40
41                 //call utility for login
42                 cy.login(thistestdata.username, thistestdata.password)
43             })
44         });
45     });
46 });


```

```
1 //use multiple fixtures for testing
2 //declare array
3 //context for us to know which test is running like tc number.
4 //we need outer describe block for the entire suite.
5 //inner describe block for the iterations.
6 const testvalueinarray=[
7
8     {
9         "name": "test1",
10        "context" :"1"
11    },
12    {
13        "name": "test2",
14        "context" :"2"
15    }
16 ]
17
18
19 //we will get run time error.
20 //Cannot read properties of undefined (reading 'name')
21 //because now this is confused which one to run so only one passes.
22 //we are not using inner describe block.
23 describe('ddt for cypress', function(){
24     testvalueinarray.forEach(function(fixturedata){
25         before(function(){
26             cy.fixture(fixturedata.name).then(function(testdata) {
27                 thistestdata=testdata;
28             })
29         })
30
31         it("login to app", function(){
32             cy.log(thistestdata.name);
33             cy.log(thistestdata.email);
34             cy.log(thistestdata.body);
35
36             //lets try to login
37             cy.visit("https://practicetestautomation.com/practice-test-login/");
38
39             //call utility for login
40             cy.login(thistestdata.username, thistestdata.password)
41         })
42     });
43 })
44
```

```
1 //use multiple fixtures for testing
2 //declare array
3 //context for us to know which test is running like tc number.
4 //we need outer describe block for the entire suite.
5 //inner describe block for the iterations.
6 const testvalueinarray=[
7
8     {
9         "name": "test1",
10        "context" :"1"
11    },
12    {
13        "name": "test2",
14        "context" :"2"
15    }
16 ]
17
18
19 //when you dont want to use inner describe block.
20 //dont use this keyword.
21 //declare global variable.
22 //assign the data to that and then use it without this.
23 describe('ddt for cypress', function(){
24     testvalueinarray.forEach(function(fixturedata){
25         lettestdata;
26         before(function(){
27             cy.fixture(fixturedata.name).then(function(data) {
28                 testdata=data;
29             })
30         })
31
32         it("login to app", function(){
33             cy.log(testdata.name);
34             cy.log(testdata.email);
35             cy.log(testdata.body);
36
37             //lets try to login
38             cy.visit("https://practicetestautomation.com/practice-test-login/");
39
40             //call utility for login
41             cy.login(testdata.username, testdata.password)
42         })
43     });
44 })
45
```

codesnap.dev

cypress auto handles all alerts.

in cypress cant switch windows or tabs directly.

```
1 //  
2 describe("handle js alerts" ,function(){  
3  
4     it("normal alert", function(){
5         cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
6         cy.contains("Click for JS Alert").click()
7
8     })
9
10    //capture text inside normal alert.
11    it("verify text inside normal alert", function(){
12        cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
13
14        cy.contains("Click for JS Alert").click()
15
16        cy.on("window:alert",function(msg){
17            expect(msg).to.be.equal("I am a JS Alert")
18        })
19
20    })
21
22
23 //capture text inside js confirm.
24 //by default cypress clicks on ok button.
25    it("click on js confirm", function(){
26        cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
27        cy.contains("Click for JS Confirm").click()
28
29        cy.on("window:confirm",function(msg){
30            expect(msg).to.be.equal("I am a JS Confirm")
31        })
32
33    })
34
35
36    //capture text inside js confirm.
37 //click on cancel button.
38    it("click on js confirm", function(){
39        cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
40        cy.contains("Click for JS Confirm").click()
41
42        cy.on("window:confirm",function(msg){
43            expect(msg).to.be.equal("I am a JS Confirm")
44
45            return false; //to click on cancel button inside js confirm.
46        })
47
48
49
50        cy.contains("You clicked: Cancel").should("be.visible")
51    })
52
53
54 //to handle prompts
55    it("click on js prompts", function(){
56        cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
57
58        //this will throw error.
59        //Timed out retrying after 4000ms: Expected to find content: 'You entered: cypress' but never did.
60        //we are clicking outside the stub so it wont work.
61
62        cy.contains("Click for JS Prompt").click()
63
64        cy.window().then(function(win){
65
66            //use stub for generating some dummy response.
67            cy.stub(win, "prompt").returns("cypress")
68        })
69
70        //assert
71        cy.contains("You entered: cypress").should("be.visible")
72    })
73
74
75 //to handle prompts inside stub
76 it.only("click on js prompts inside stub", function(){
77     cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
78
79     //this will throw error.
```

```
1 //  
2 describe("handle js alerts" ,function(){  
3  
4     it("normal alert", function(){  
5         cy.visit("https://the-internet.herokuapp.com/javascript_alerts")  
6         cy.contains("Click for JS Alert").click()  
7     })  
8  
9     //capture text inside normal alert.  
10    it("verify text inside normal alert", function(){  
11        cy.visit("https://the-internet.herokuapp.com/javascript_alerts")  
12        cy.contains("Click for JS Alert").click()  
13  
14        cy.on("window:alert",function(msg){  
15            expect(msg).to.be.equal("I am a JS Alert")  
16            })  
17        })  
18  
19    })  
20  
21
```

```
1 //capture text inside js confirm.
2 //by default cypress clicks on ok button.
3 it("click on js confirm", function(){
4     cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
5     cy.contains("Click for JS Confirm").click()
6
7     cy.on("window:confirm",function(msg){
8         expect(msg).to.be.equal("I am a JS Confirm")
9     })
10
11 })
12
13
14 //capture text inside js confirm.
15 //click on cancel button.
16 it("click on js confirm", function(){
17     cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
18     cy.contains("Click for JS Confirm").click()
19
20     cy.on("window:confirm",function(msg){
21         expect(msg).to.be.equal("I am a JS Confirm")
22
23         return false; //to click on cancel button inside js confirm.
24     })
25
26
27
28     cy.contains("You clicked: Cancel").should("be.visible")
29 })
```

```
1 //to handle prompts
2   it("click on js prompts", function(){
3     cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
4
5     //this will throw error.
6     //Timed out retrying after 4000ms: Expected to find content: 'You entered: cypress' but never did.
7     //we are clicking outside the stub so it wont work.
8
9     cy.contains("Click for JS Prompt").click()
10
11    cy.window().then(function(win){
12
13      //use stub for generating some dummy response.
14      cy.stub(win, "prompt").returns("cypress")
15    })
16
17      //assert
18      cy.contains("You entered: cypress").should("be.visible")
19    })
20
21
22 //to handle prompts inside stub
23 it.only("click on js prompts inside stub", function(){
24   cy.visit("https://the-internet.herokuapp.com/javascript_alerts")
25
26   //this will throw error.
27   //Timed out retrying after 4000ms: Expected to find content: 'You entered: cypress' but never did.
28   //we are clicking outside the stub so it wont work.
29
30
31   cy.window().then(function(win){
32     cy.contains("Click for JS Prompt").click()
33     //use stub for generating some dummy response.
34     cy.stub(win, "prompt").returns("cypress")
35   })
36
37
38      //assert
39      cy.contains("You entered: cypress").should("be.visible")
40    })
41
42 })
```

codesnap.dev

multiple windows or tabs not supported in cypress. we need to manually remove the properties before working.

```
1  //  
2  describe("test multi tabs or windows", function(){  
3      it("check for multi tabs or windows", function(){  
4  
5          //visit page  
6          cy.visit("https://the-internet.herokuapp.com/windows")  
7  
8          //first remove attribute which opens new tab and then perform click.  
9          cy.get("a[href='/windows/new']").invoke("removeAttr", "target").click()  
10     })  
11  
12  
13  
14     //switch to new tab and perform some operation on that window  
15     it.only("switch to new tab and perform operation", function(){  
16  
17         //visit page  
18         cy.visit("https://ineuron-courses.vercel.app/login")  
19  
20         //login  
21         cy.get("#email1").type("ineuron@ineuron.ai")  
22         cy.get("#password1").type("ineuron")  
23         cy.get("button[type='submit']").click()  
24  
25         cy.contains("Manage").realHover();  
26  
27         cy.wait(3000);  
28  
29         cy.contains("Manage Categories").invoke("removeAttr", "target").click({force:true})  
30  
31         //lets validate on new window  
32         cy.window().then(function(win){  
33             cy.contains("Add New Category").click();  
34             cy.stub(win, "prompt").returns("playwright")  
35         })  
36  
37         cy.contains("playwright").should("be.visible")  
38     })  
39 })
```

codesnap.dev

frames-

not supported.

again install and add in the support file.

how to do js way-

```
> document.getElementsByTagName("iframe")
< HTMLCollection [frame@mce_0_ifr,tox-edit-area__frame, mce_0_ifr: frame@mce_0_ifr,tox-edit-area__frame]
> document.getElementsByTagName("iframe")[0]
< <iframe id="mce_0_ifr" frameborder="0" allowtransparency="true" title="Rich Text Area" class="tox-edit-area__iframe"></iframe>
> document.getElementsByTagName("iframe")[0].contentDocument
< #document
<!DOCTYPE html>
<html>
<head>
<body id="tinyMCE" class="mce-content-body " data-id="mce_0" aria-label="Rich Text Area. Press ALT-0 for help." contenteditable="true" spellcheck="false" data-new-gr-c-s-loaded="14.1080.0" data-new-gr-c-s-check-loaded="1
4.1080.0" data-gr-ext-installed></body>
<grammarly-desktop-integration data-grammarly-shadow-root="true"></grammarly-desktop-integration>
</html>
```

```
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p")
< ▼<p>
  <br data-mce-bogus="1">
</p>
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p").value="Mukesh"
< 'Mukesh'
```

```
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p")
< ▼<p>
  <br data-mce-bogus="1">
</p>
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p").value="Mukesh"
< 'Mukesh'
```

```
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p")
< ▼<p>
  <br data-mce-bogus="1">
</p>
> document.getElementsByTagName("iframe")[0].contentDocument.querySelector("p").value="Mukesh"
< 'Mukesh'
```

```
1 //  
2 //get into the frame first.  
3 //using its for multiple web elements.  
4 //get the frame area then use wrap to make it usable in cypress.  
5 //then use cypress commands.  
6  
7 describe("handle frames without cypress",function(){  
8     it("handle frame", function(){  
9  
10        //visit url  
11        cy.visit("https://the-internet.herokuapp.com/iframe")  
12  
13        cy.get("iframe")  
14        // .its("0.contentDocument.body") //go till the tag where the frame is present OR  
15        .its("0.contentDocument")  
16        .its("body")  
17        .then(function(body){  
18            cy.wrap(body).clear().type("cypress")  
19        })  
20    })  
21  
22  
23  
24  
25    it.only("handle frame", function(){  
26  
27        //visit url  
28        cy.visit("https://rahulshettyacademy.com/AutomationPractice/")  
29  
30        cy.get("iframe")  
31        // .its("0.contentDocument.body") //go till the tag where the frame is present OR  
32        .its("0.contentDocument")  
33        .its("body")  
34        .then(function(body){  
35  
36            //You're invoking cy.wrap() immediately, which causes Cypress to pass undefined to  
37            // .then() rather than passing a function. Cypress expects a function in  
38            // .then() – not the result of a function call.  
39            cy.wrap(body).  
40            find(".theme-btn.register-btn")  
41            .click({force:true})  
42        })  
43  
44    })  
45})  
46})
```