

```
new Go Run Terminal Window Help
DeleteUser.js — CypressAPITraining
JS PostUser.js JS PutUser.js M JS weatherapi.js commands.js JS GetUser.js M JS DeleteUser.js
cypress > integration > APITests > JS DeleteUser.js > describe('Delete user request') callback > it.only('create user test')
2 const dataJson = require('../../fixtures/createuser')
3
4 describe('Delete user request', () => {
5   let accessToken = '007526d9efdbc07e084ff7a6d4cfcc90588fbe20641c00faebf45a7f3b2eaf33'
6   let randomText = ''
7   let testEmail = ''
8   it.only('create user test', () => {
9
10
```

```
cypress > integration > APITests > JS DeleteUser.js > describe('Delete user request') callback > it.only('create user test')
6 let randomText = ''
7 let testEmail = ''
8 it.only('create user test', () => {
9   cy.request('POST', 'http://localhost:8080/api/v1/users', {
10     //1. create user (POST)
11     cy.request({
12       method: 'POST',
13       url: 'https://gorest.co.in/public/v1/users',
14       headers: {
```

```
13       url: 'https://gorest.co.in/public/v1/users',
14       headers: {
15         'Authorization': 'Bearer ' + accessToken
16       },
17       body: {
18         "name": "Test Automation Cypress User",
19         "gender": "male",
20         "email": "naveenCypressuser@gmail.com",
21         "status": "active"
22       }
23     })
24     function(this: Cypress.ObjectLike, res: Cypress.ResponseLike) {
```

```
23
24   }).then((res) => {
25     cy.log(JSON.stringify(res))
26     expect(res.status).to.eq(201)
27     expect(res.body.data).has.property('email', 'naveenCypressuser@gmail.com')
28     expect(res.body.data).has.property('name', 'Test Automation Cypress User')
29   }).then((res) => {
```

```
28     expect(res.body.data).has.property('name','Test Automation Cypress Use
29   }).then((res) =>{
30     const userId = res.body.data.id
31     cy.log("user id is: " + userId)
32     //2. delete user (DELETE)
33     cy.request({
34       method: 'DELETE',
35       url: 'https://gorest.co.in/public/v1/users/'+userId,
36       headers: {
37         'Authorization': 'Bearer ' + accessToken
38       }
39     }).then((res)=>{
40       expect(res.status).to.eq(204)
41     })
```