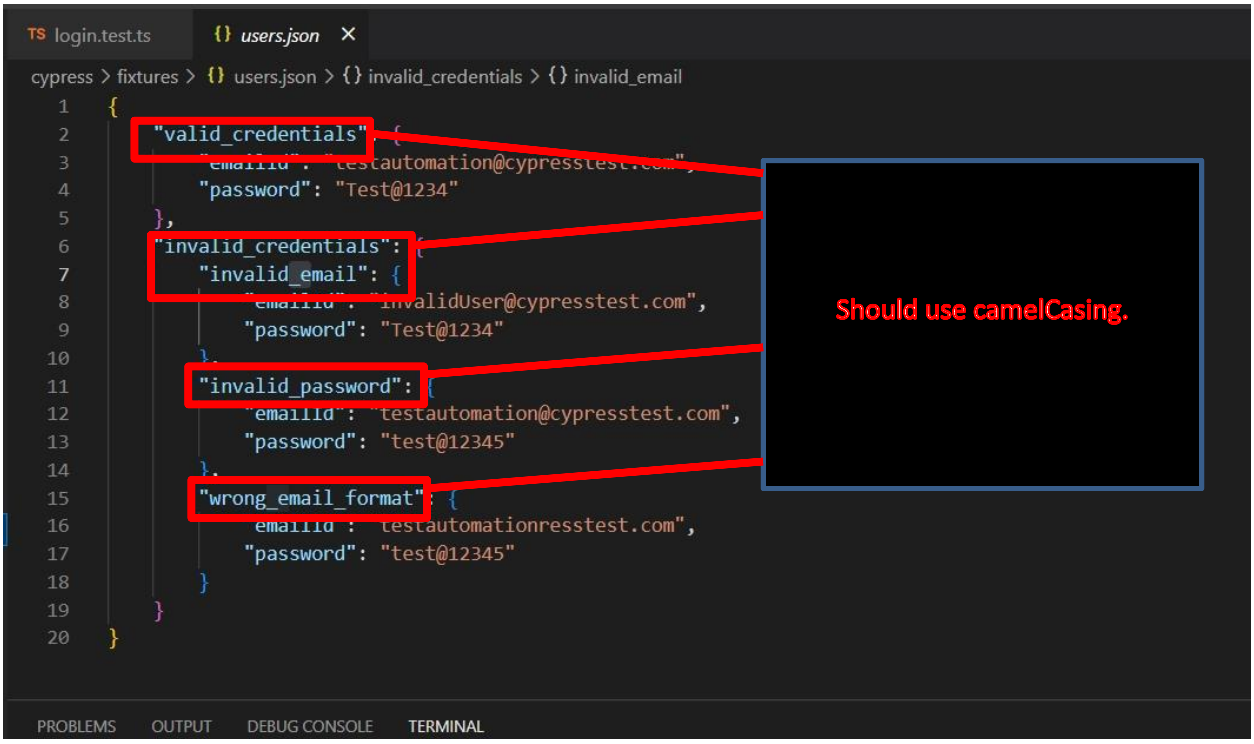
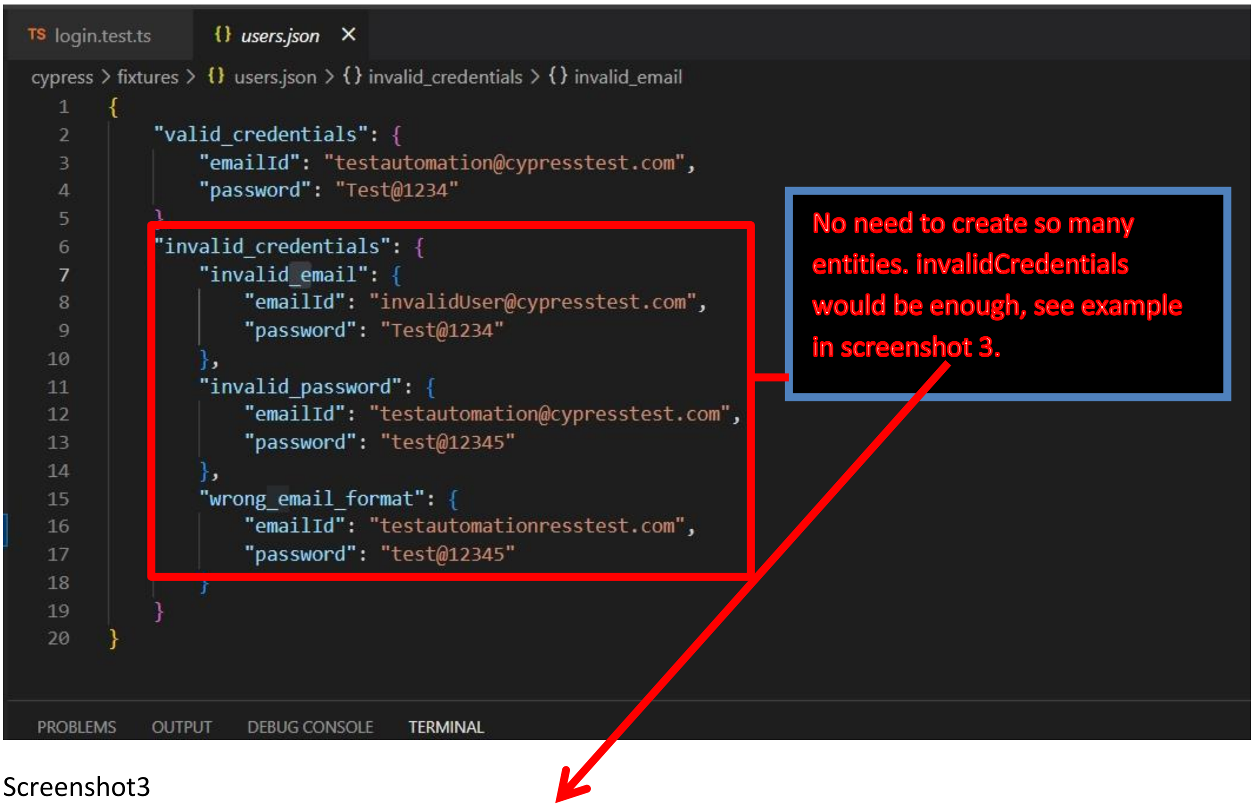
Screenshot1


TS login.test.ts
{} users.json ×
cypress > fixtures > {} users.json > {} invalid_credentials > {} invalid_email
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
{
    "valid_credentials": {
        "emailId": "testautomation@cypresstest.com",
        "password": "Test@1234"
    },
    "invalid_credentials": {
        "invalid_email": {
            "emailId": "invalidUser@cypresstest.com",
            "password": "Test@1234"
        },
        "invalid_password": {
            "emailId": "testautomation@cypresstest.com",
            "password": "test@12345"
        },
        "wrong_email_format": {
            "emailId": "testautomationresstest.com",
            "password": "test@12345"
        }
    }
}
```

Should use camelCasing.

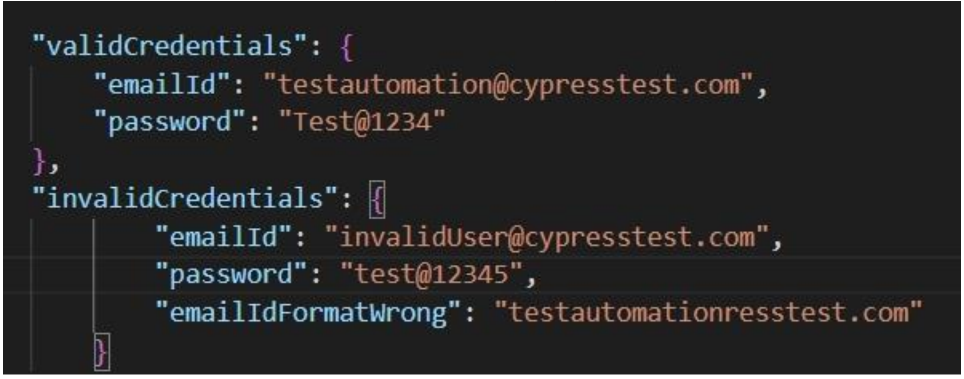
Screenshot2


TS login.test.ts
{} users.json ×
cypress > fixtures > {} users.json > {} invalid_credentials > {} invalid_email
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
{
    "valid_credentials": {
        "emailId": "testautomation@cypresstest.com",
        "password": "Test@1234"
    },
    "invalid_credentials": {
        "invalid_email": {
            "emailId": "invalidUser@cypresstest.com",
            "password": "Test@1234"
        },
        "invalid_password": {
            "emailId": "testautomation@cypresstest.com",
            "password": "test@12345"
        },
        "wrong_email_format": {
            "emailId": "testautomationresstest.com",
            "password": "test@12345"
        }
    }
}
```

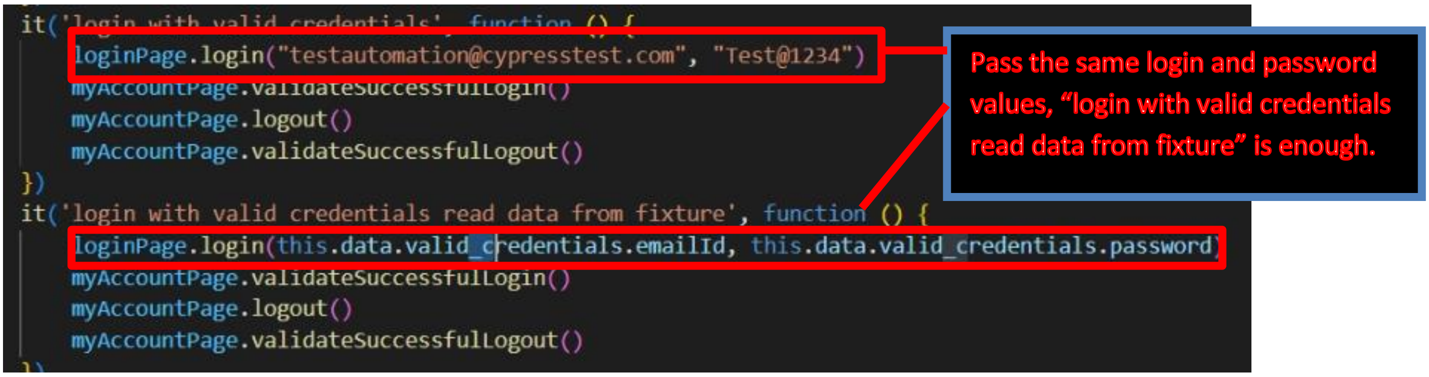
No need to create so many entities. invalidCredentials would be enough, see example in screenshot 3.

Screenshot3



```
"validCredentials": {
    "emailId": "testautomation@cypresstest.com",
    "password": "Test@1234"
},
"invalidCredentials": {
        "emailId": "invalidUser@cypresstest.com",
        "password": "test@12345",
        "emailIdFormatWrong": "testautomationresstest.com"
    }
```

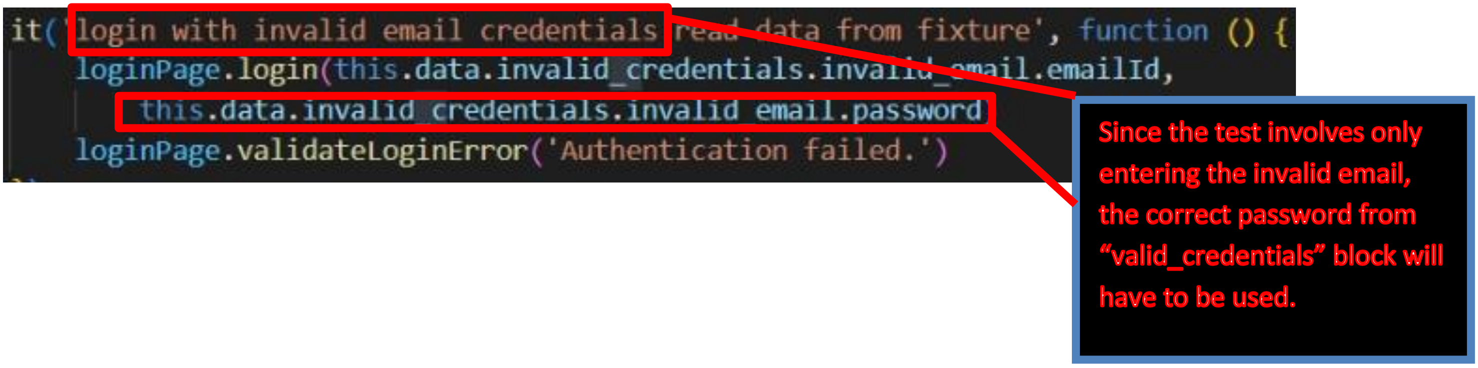
Screenshot4



```
it('login with valid credentials', function () {
    loginPage.login("testautomation@cypresstest.com", "Test@1234")
    myAccountPage.validateSuccessfulLogin()
    myAccountPage.logout()
    myAccountPage.validateSuccessfulLogout()
})
it('login with valid credentials read data from fixture', function () {
    loginPage.login(this.data.valid_credentials.emailId, this.data.valid_credentials.password)
    myAccountPage.validateSuccessfulLogin()
    myAccountPage.logout()
    myAccountPage.validateSuccessfulLogout()
```

Pass the same login and password values, “login with valid credentials read data from fixture” is enough.

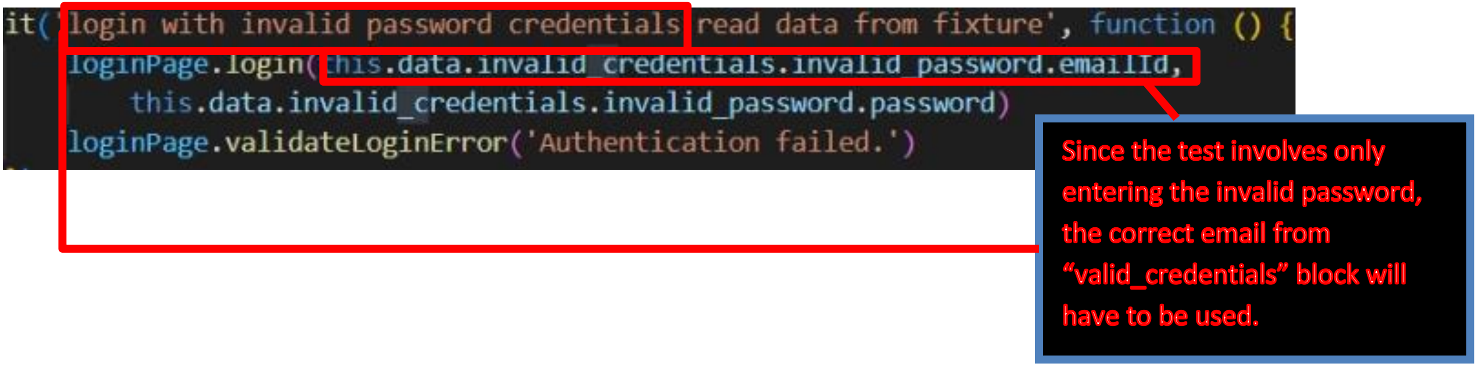
Screenshot5



```
it('login with invalid email credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.invalid_email.emailId,
        this.data.invalid_credentials.invalid_email.password
    loginPage.validateLoginError('Authentication failed.')
```

Since the test involves only entering the invalid email, the correct password from “valid_credentials” block will have to be used.

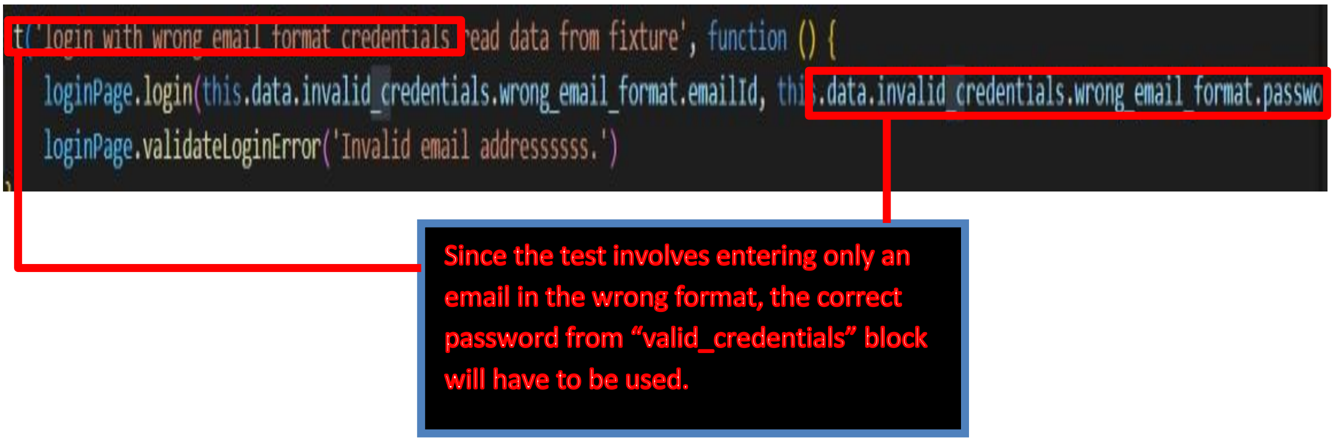
Screenshot6



```
it('login with invalid password credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.invalid_password.emailId,
        this.data.invalid_credentials.invalid_password.password)
    loginPage.validateLoginError('Authentication failed.')
```

Since the test involves only entering the invalid password, the correct email from “valid_credentials” block will have to be used.

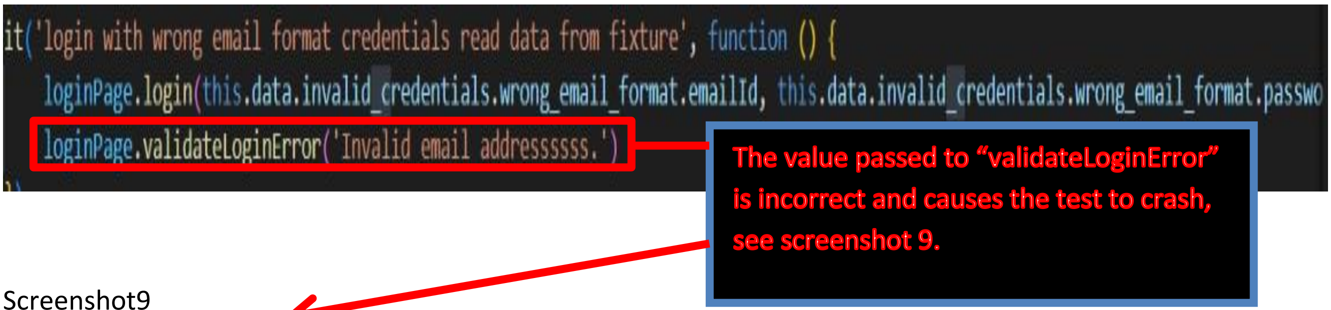
Screenshot7



```
it('login with wrong email format credentials read data from fixture', function () {
    loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.passwo
    loginPage.validateLoginError('Invalid email addresssss.')
```

Since the test involves entering only an email in the wrong format, the correct password from “valid_credentials” block will have to be used.

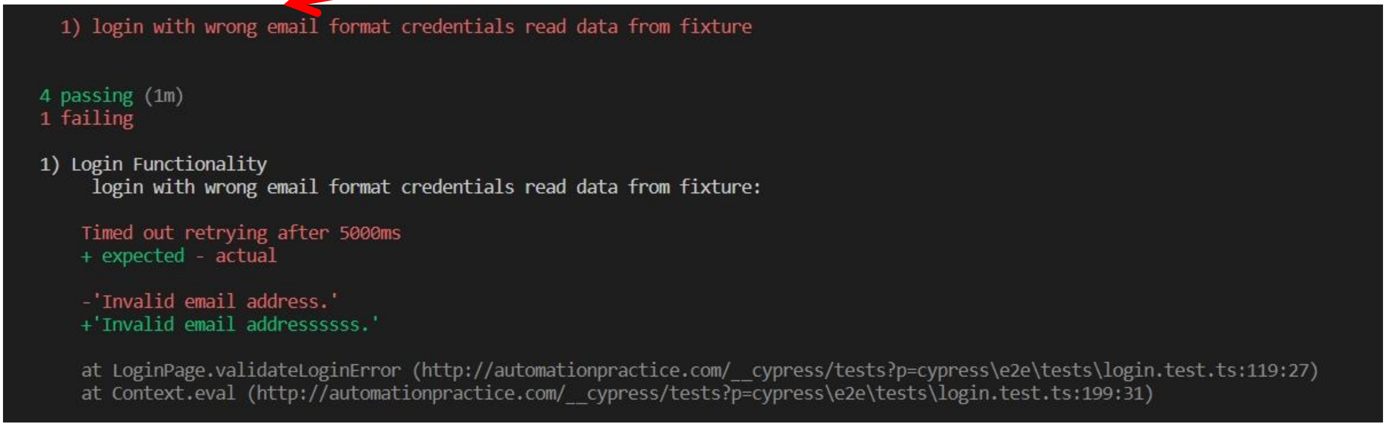
Screenshot8



```
it('login with wrong email format credentials read data from fixture', function () {
   loginPage.login(this.data.invalid_credentials.wrong_email_format.emailId, this.data.invalid_credentials.wrong_email_format.passwo
   loginPage.validateLoginError('Invalid email addresssss.')
})
```

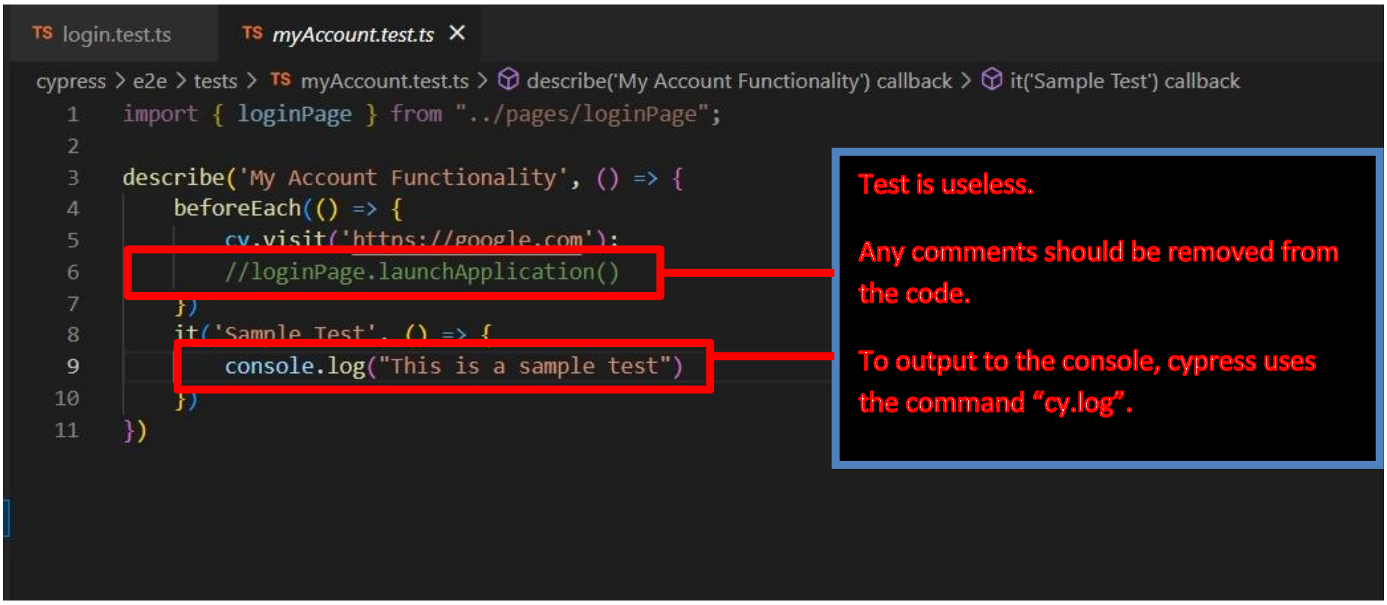
The value passed to "validateLoginError" is incorrect and causes the test to crash, see screenshot 9.

Screenshot9



```
  1) login with wrong email format credentials read data from fixture


  4 passing (1m)
  1 failing

  1) Login Functionality
       login with wrong email format credentials read data from fixture:

      Timed out retrying after 5000ms
      + expected - actual

      -'Invalid email address.'
      +'Invalid email addresssss.'

      at LoginPage.validateLoginError (http://automationpractice.com/__cypress/tests?p=cypress\e2e\tests\login.test.ts:119:27)
      at Context.eval (http://automationpractice.com/__cypress/tests?p=cypress\e2e\tests\login.test.ts:199:31)
```

Screenshot10



TS login.test.ts
TS myAccount.test.ts ✕
cypress > e2e > tests > TS myAccount.test.ts > describe('My Account Functionality') callback > it('Sample Test') callback

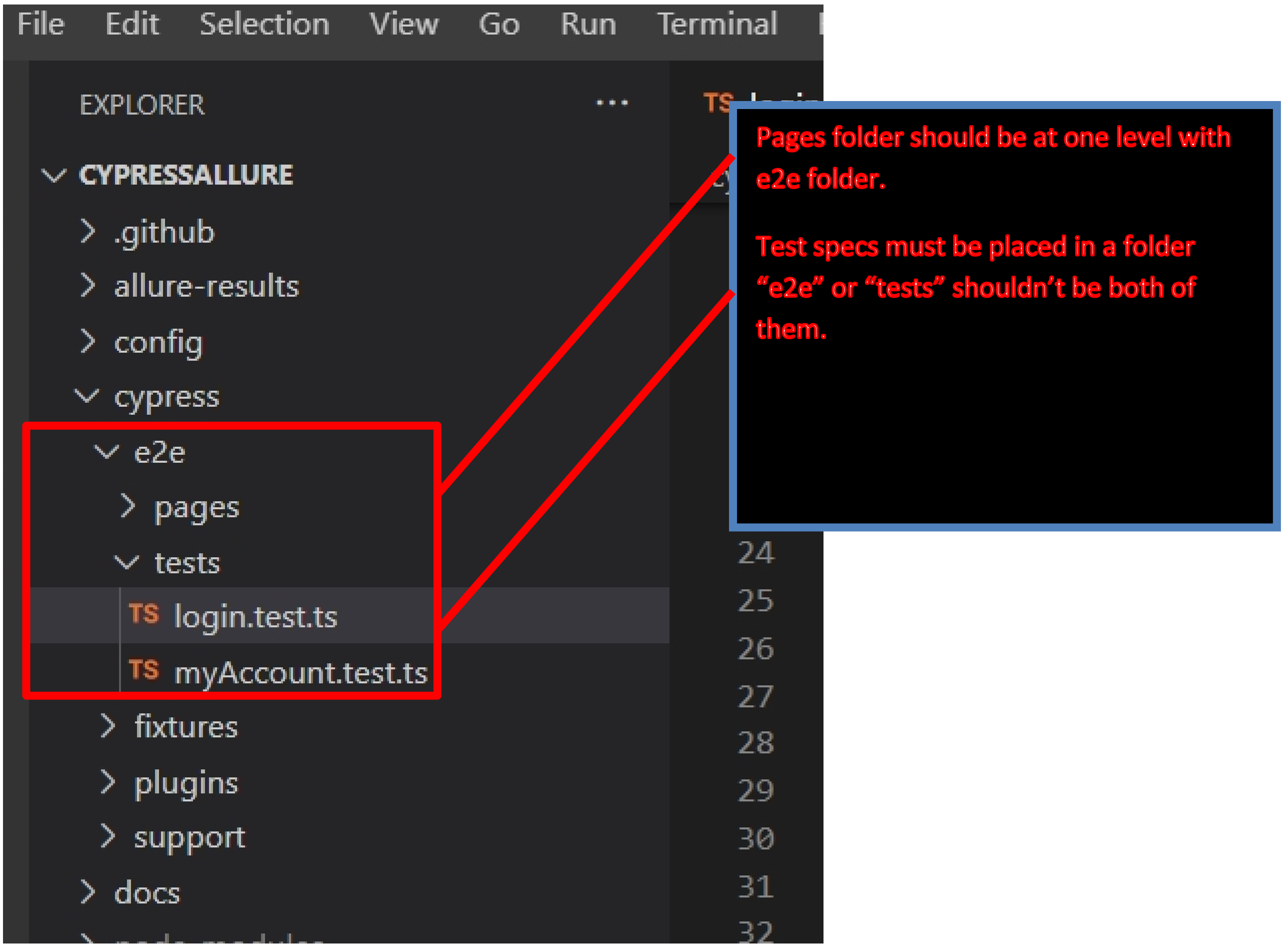
```
import { loginPage } from "../pages/loginPage";

describe('My Account Functionality', () => {
    beforeEach(() => {
        cy.visit('https://google.com');
        //loginPage.launchApplication()
    })
    it('Sample Test', () => {
        console.log("This is a sample test")
    })
})
```

Test is useless.

Any comments should be removed from the code.

To output to the console, cypress uses the command "cy.log".

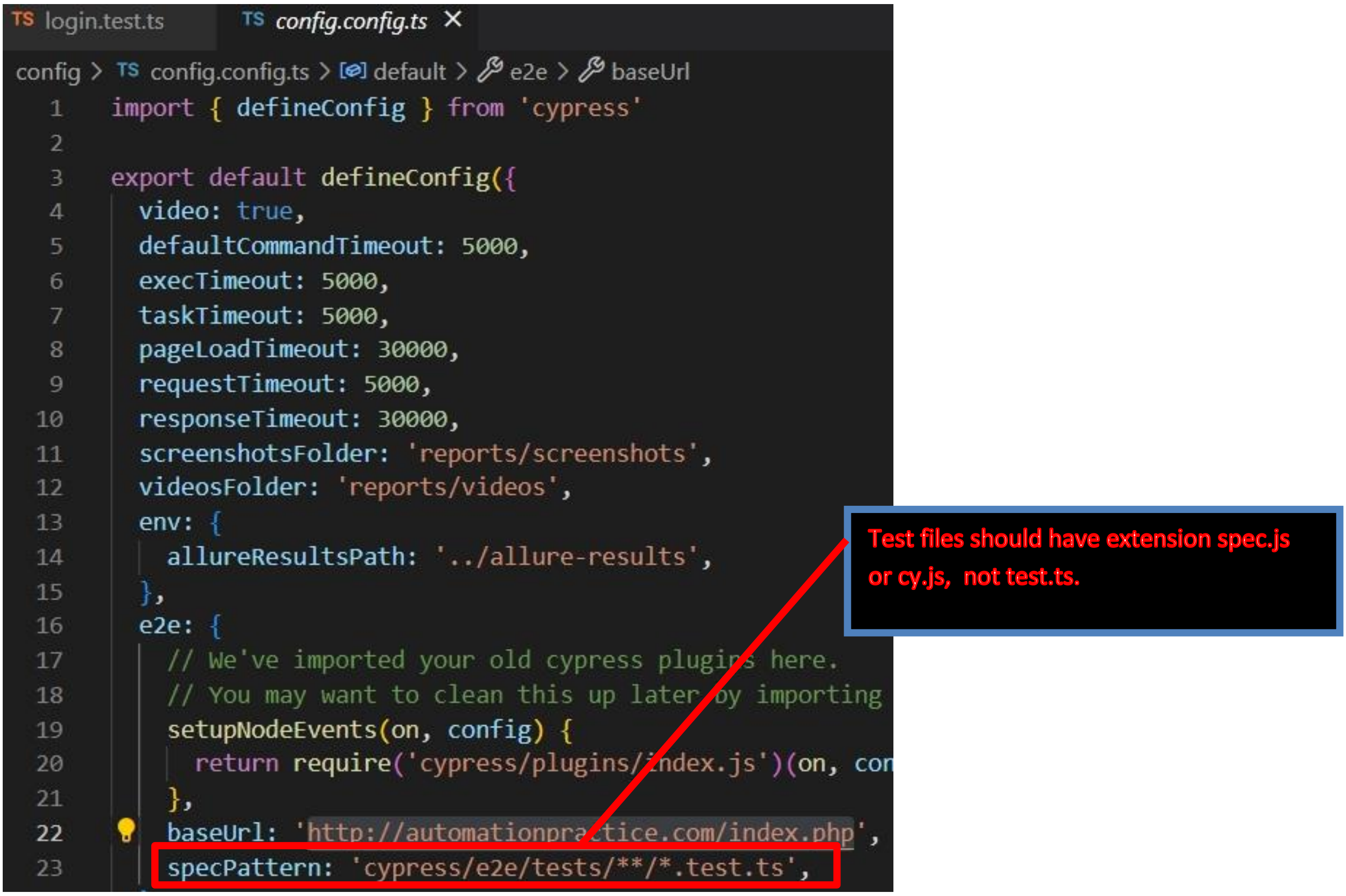
Screenshot10



Pages folder should be at one level with e2e folder.

Test specs must be placed in a folder “e2e” or “tests” shouldn’t be both of them.

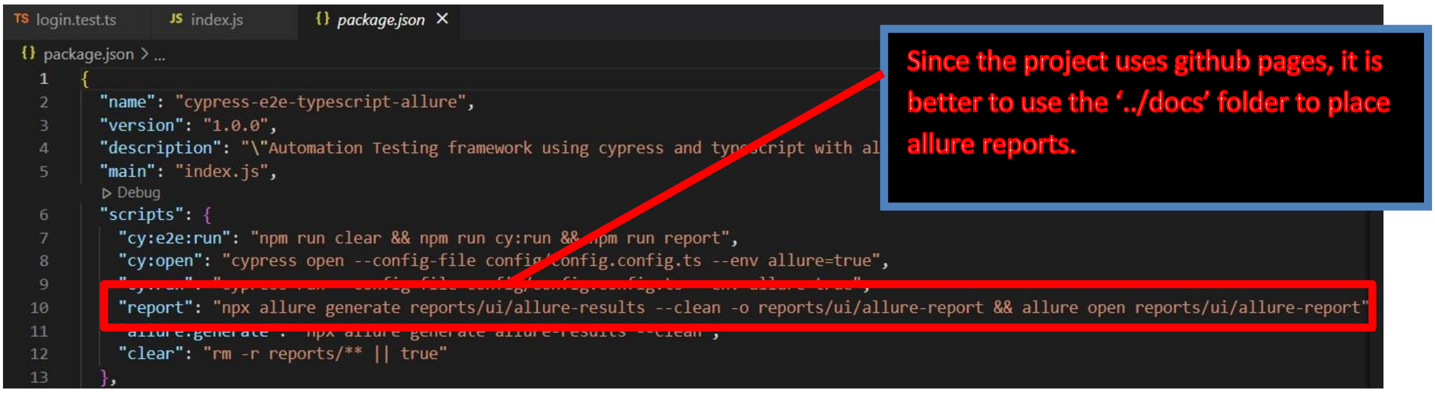
Screenshot11



```
import { defineConfig } from 'cypress'

export default defineConfig({
  video: true,
  defaultCommandTimeout: 5000,
  execTimeout: 5000,
  taskTimeout: 5000,
  pageLoadTimeout: 30000,
  requestTimeout: 5000,
  responseTimeout: 30000,
  screenshotsFolder: 'reports/screenshots',
  videosFolder: 'reports/videos',
  env: {
    allureResultsPath: '../allure-results',
  },
  e2e: {
    // We've imported your old cypress plugins here.
    // You may want to clean this up later by importing
    setupNodeEvents(on, config) {
      return require('cypress/plugins/index.js')(on, con
    },
    baseUrl: 'http://automationpractice.com/index.php',
    specPattern: 'cypress/e2e/tests/**/*.test.ts',
```

Test files should have extension spec.js or cy.js, not test.ts.

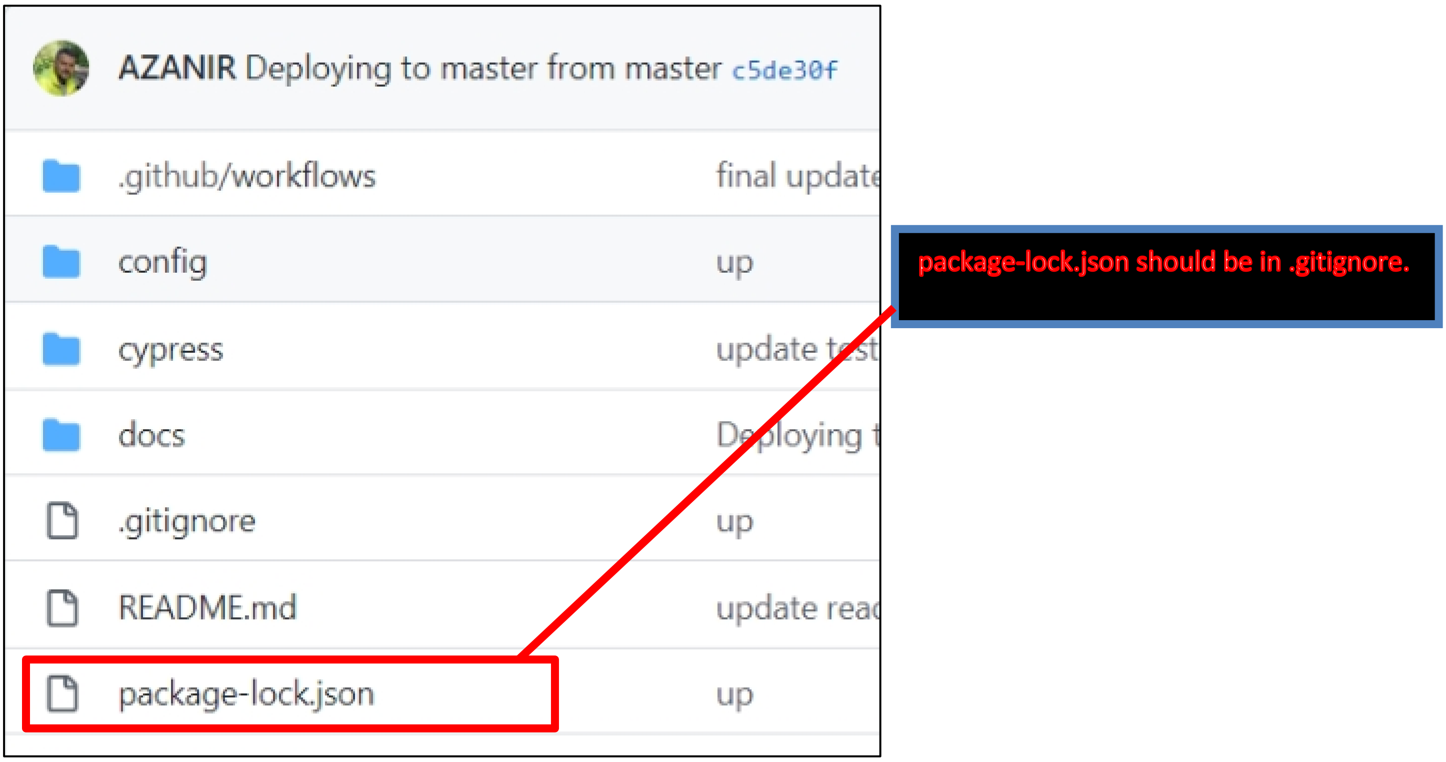
Screenshot12



```
{
  "name": "cypress-e2e-typescript-allure",
  "version": "1.0.0",
  "description": "\"Automation Testing framework using cypress and typescript with al
  "main": "index.js",
  "scripts": {
    "cy:e2e:run": "npm run clear && npm run cy:run && npm run report",
    "cy:open": "cypress open --config-file config/config.config.ts --env allure=true",
    "cy:run": "cypress run --config-file config/config.config.ts --env allure=true",
    "report": "npx allure generate reports/ui/allure-results --clean -o reports/ui/allure-report && allure open reports/ui/allure-report"
    "allure:generate": "npx allure generate allure-results --clean",
    "clear": "rm -r reports/** || true"
  },
```

Since the project uses github pages, it is better to use the '../docs' folder to place allure reports.

Screenshot13

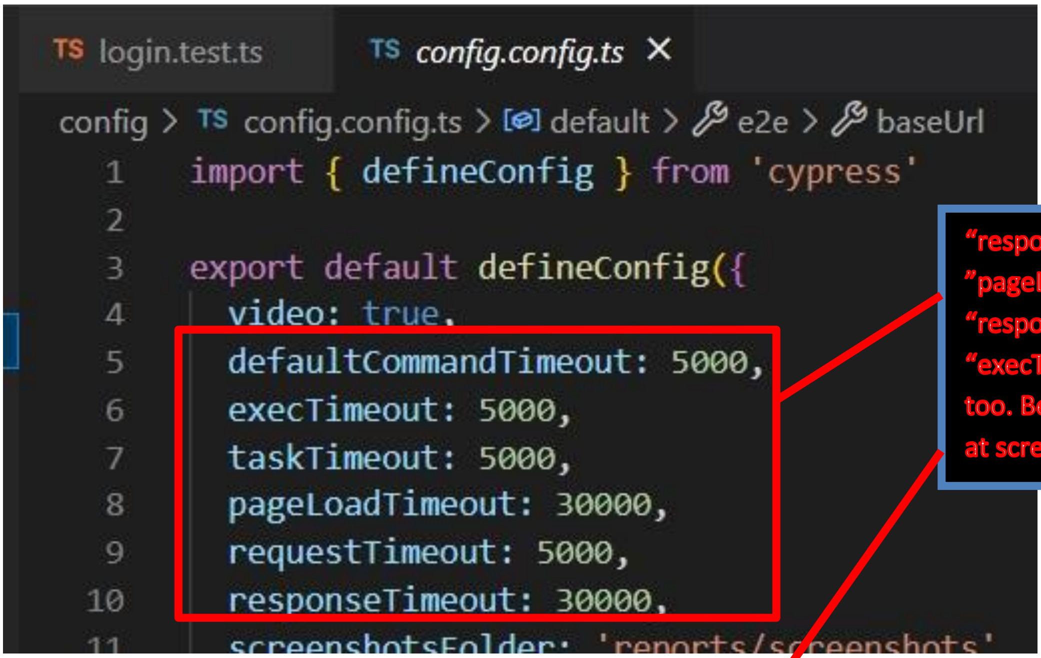


AZANIR Deploying to master from master c5de30f

.github/workflows	final update
config	up
cypress	update test
docs	Deploying t
.gitignore	up
README.md	update read
package-lock.json	up

package-lock.json should be in .gitignore.

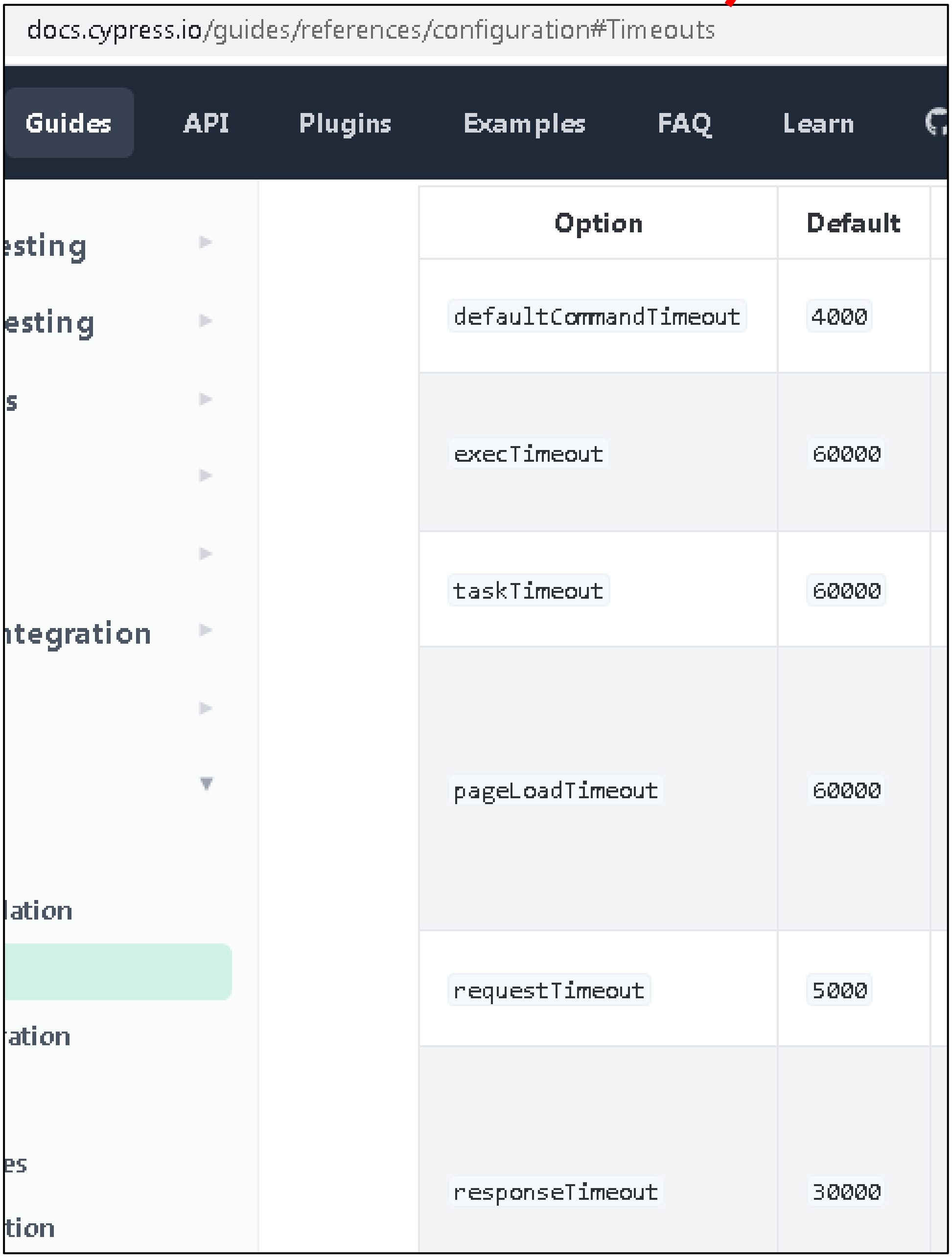
Screenshot14



```
import { defineConfig } from 'cypress'

export default defineConfig({
  video: true,
  defaultCommandTimeout: 5000,
  execTimeout: 5000,
  taskTimeout: 5000,
  pageLoadTimeout: 30000,
  requestTimeout: 5000,
  responseTimeout: 30000,
  screenshotsFolder: 'reports/screenshots'
```

"responseTimeout" is equal "pageLoadTimeout" – it's wrong, "responseTimeout" less than "execTimeout" & "taskTimeout" wrong too. Better to use default timeouts(look at screenshot 15)

Screenshot15



Option	Default
defaultCommandTimeout	4000
execTimeout	60000
taskTimeout	60000
pageLoadTimeout	60000
requestTimeout	5000
responseTimeout	30000