|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Cues**  NoteGem Horizontal Line        why should you use cypress?            how do you install & run cypress?                          how does cypress work?                      how long does cypress wait by default when making assertions?                 what does cy.task do?                                                                                      how do you use parallelization in cypress? | **Notes**  NoteGem Horizontal Line    docs.cypress.io/API  completely tech-stack agnostic    **why cypress**  non-technical  growing in popularity/adoption  growing demand for testing skills & cypress    **commands:**  yarn add cypress / npm -i -D cypress  npx cypress open    very first time, cypress with scaffold directory  can open specs in browser  takes snapshots of app after every test    cy.<command>    cy.get('button').click().should('have.class', 'active')    cy.request('/users/1').its('body').should('deep.eql', {name: 'Amir'})    is a chaining API  handles subject passed through chain  cypress won't imperatively execute commands  will make to-do list and run everything in order you asked for  runs in deterministic manner    it('send email with contact form'), ()=>{  cy.get('#name-input').type('Amir')  cy.get('form').submit()  cy.get('#success\_msg').should('be.visible')  })    waits when making assertions (four seconds by default**)**    it('sign up and log in user'), ()=>{  cy.visit('urlsite')  cy.get('input[name="email"]').type('amir@cypress.io')  cy.get('#signup-button').click()  cy.location('pathname').should('eq', '/login')  })    **cy.task** = run javascript at system level outside of browser in cypress/plugins/index.js    const {clearDatabase} = require('../../server/db') THEN refer to it    module.exports = (on, config) =>{  on('task'{  'clear.db' ()=>{  return clearDatabase();  }})}    cy.task('clear:db')    cy.task('seed:db', seedData.data)    custom: cy.login('email', 'password')    cypress.commands.add('login', (email, password)=>{  cy.get('input[name="email"]).type(email)  cy.get('input[name="password"]).type(password)  cy.get('#login-button').click()  })    cypress.commands.add('login', (email, password)=>{  return cy.window().then(win=>{  return win.app.$store.dispatch('login',{  email: 'amir@cypress.io',  password: '1234'  })  })  })    cy.route('GET', '/tweets\*', 'fixture: tweets').as('tweets') // is a label    cy.get('#hashtags').type('javascript{enter}').type('cypressio{enter}')    cy.window().then(win=>{  cy.wait('@tweets') //label  .its('response.body.tweets') //response  .should('have.length', win.app.$store.state.tweets.length)  })  })    npx cypress run (-record) (-record --parallel)    **parallel** = optimizes CI usage with parallelization    **beforeEach(()=>{})** = runs before each individual test | |
| |  | | --- | | **Summary** | |