CodeceptJS

A field manual for end-to-end testing — locators, actions, assertions & page objects on a single sheet.

№ 01
REFERENCE POSTER
A4 · CODECEPT.IO

01 TEST STRUCTURE

One Feature per file — I is the actor

```
Feature('User Login')
Before(({ I }) => I.amOnPage('/'))
Scenario('login works', ({ I }) => {
  I.click('Login')
  I.fillField('Email', 'user@x.com')
  I.fillField('Password', secret('s3cret'))
  I.see('Welcome')
  I.seeInCurrentUrl('/dashboard')
})
```

02 ASYNC & AWAIT

Actions auto-chain

```
// queued — no await
I.amOnPage('/')
I.click('Login')

// await grab* & page objects
const pw = await I.grabTextFrom('#pw')
I.fillField('Password', pw)

await loginPage.login(user)
```

03 CONFIG & HELPERS

Playwright · WebDriver · Puppeteer · Appium

```
// codecept.conf.js
export const config = {
  helpers: { Playwright: {
    url: 'http://localhost:3000',
    browser: 'chromium',
  } },
  include: { loginPage: './pages/Login.js' },
  tests: './**/*_test.js',
}
```

04 LOCATORS

Strict objects beat fuzzy strings

```
I.click({ role: 'button', name: 'Save' })  // ARIA — preferred
I.click({ css: '.btn-save' })              // CSS
I.click({ xpath: '//button[1]' })          // XPath
I.click('#signup')                         // id shortcut
I.click('~login')                          // aria-label
I.click('Sign In')                         // semantic (fuzzy)
I.click('Delete', '.toolbar')              // + context
I.click('a', step.opts({ elementIndex: 2 }))   // Nth match
I.click('a', step.opts({ exact: true }))     // strict
```

05 LOCATE() BUILDER

XPath only — not ARIA

```
// chain → single XPath
locate('tr')
  .withDescendant(locate('td').withText('Acme'))
  .find('button').withText('Edit')
// other methods
locate('input').withAttr({ type: 'email' })
locate('a').inside('nav').first()
locate('li').withClassAttr('active').at(-1)
locate('form').withChild('select')
```

06 PAGE OBJECTS

Classes with lifecycle hooks

```
const { I } = inject()
class LoginPage {
  sendForm(email, pw) {
    I.fillField('Email', email)
    I.fillField('Password', pw)
    I.click('Submit')
  }
  _before() { I.amOnPage('/login') }
}
```

07 CLICKS & NAVIGATION

Pointer & keyboard

```
I.amOnPage('/about')
I.click('Login')
I.click('Save', '.toolbar')
I.click({ role: 'button', name: 'OK' })
I.forceClick('.hidden')
I.doubleClick('.row')
I.rightClick('.ctx')
I.dragAndDrop('#src', '#dst')
I.pressKey('Enter')
```

08 FORMS

Inputs · selects · files

```
I.fillField('Email', 'user@test.com')
I.fillField('Password', secret('s3cret!'))
I.fillField('Email', email, '#signup')  // scope
I.appendField('Notes', ' — urgent')
I.clearField('#search')
I.selectOption('Country', 'Ukraine')
I.selectOption('Tags', ['urgent', 'review'])
I.checkOption('I Agree')
I.attachFile('#upload', 'invoice.pdf')
```

09 ASSERTIONS

Every see has a dontSee* counterpart*

```
I.see('Welcome')
I.see('Error', '.alert')
I.dontSee('Loading')
I.seeElement({ role: 'button', name: 'Go' })
I.dontSeeElement('.spinner')
I.seeInCurrentUrl('/dashboard')
I.seeInField('Email', 'user@test.com')
I.seeNumberOfElements('.item', 5)
I.seeCookie('session')
```

10 GRABBERS

Always require await

```
const t   = await I.grabTitle()
const txt = await I.grabTextFrom('.name')
const arr = await I.grabTextFromAll('.i')
const v   = await I.grabValueFrom('#q')
const h   = await I.grabAttributeFrom('a', 'href')
const url = await I.grabCurrentUrl()
const ck  = await I.grabCookie('session')
const el  = await I.grabWebElement('.btn')
const els = await I.grabWebElements('.row')
```

11 WAITING

Explicit state sync

```
I.waitForVisible('.modal', 5)
I.waitForInvisible('.spinner')
I.waitForElement('.r li', 10)
I.waitForText('Done', 5, '.alert')
I.waitForClickable('.btn')
I.waitForEnabled('#submit')
I.waitForFunction(() => ready)
I.waitInUrl('/ok')
I.wait(2)         // seconds
```

12 RUNNING TESTS

CLI · filters · workers · generators

```
# basic · filter
npx codeceptjs run
npx codeceptjs run tests/login_test.js
npx codeceptjs run --grep "@smoke"
npx codeceptjs run --grep "checkout"
npx codeceptjs run --debug
npx codeceptjs run-workers 3      # parallel
npx codeceptjs run -c codecept.ci.conf.js
```