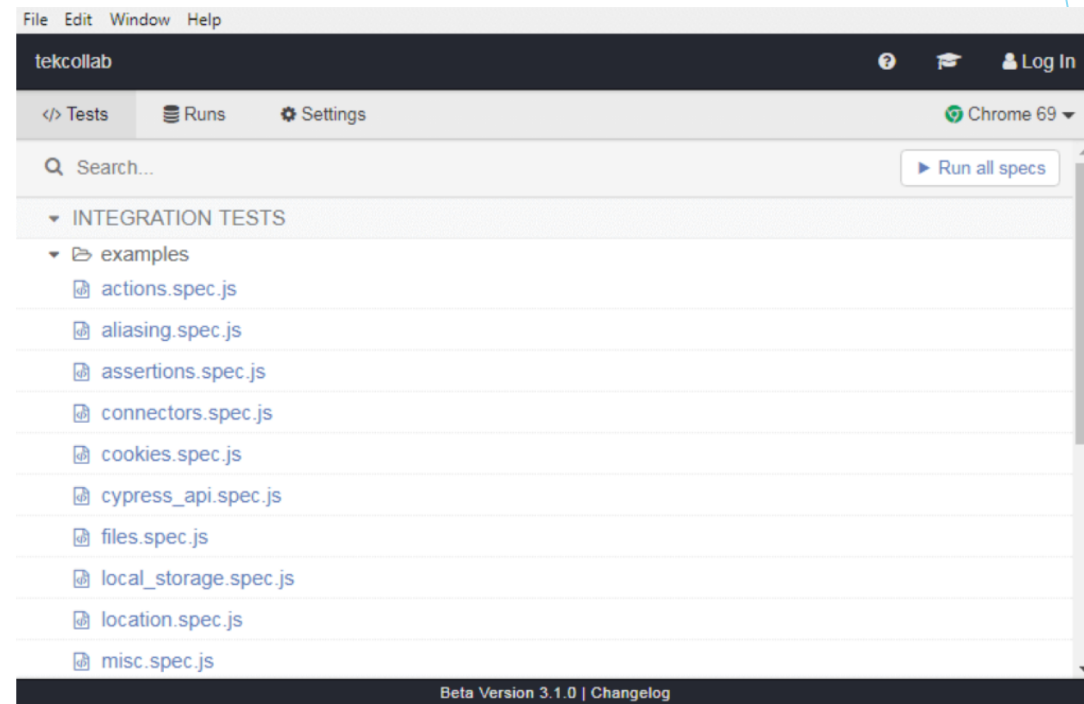


The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the left and right sides of the frame, creating a modern, dynamic feel. The central area is a plain, light grayish-white, providing a high-contrast space for the text.

Test e2e Wbe Cypress

Installation

- ▶ `npm install cypress --save-dev`
- ▶ Attention ne fonctionne que sous chrome
- ▶ Lancement
 - ▶ `./node_modules/.bin/cypress open`
 - ▶ Ouverture de la fenêtre des test
 - ▶ Il suffit de cliquer sur le test pour le lancer.



Premier test

```
describe('My First Test', () => {  
  it('Visits the Kitchen Sink', () => {  
    cy.visit('https://example.cypress.io')  
  })  
})
```

```
describe('My First Test', () => {  
  it('Does not do much!', () => {  
    expect(true).to.equal(true)  
  })  
})
```

▼ My First Test

✓ Does not do much!

▼ TEST

1

ASSERT

expected true to equal true

Action

```
describe('My First Test', () => {  
  it('clicking "type" navigates to a new url', () => {  
    cy.visit('https://example.cypress.io')  
  
    cy.contains('type').click()  
  
    // Should be on a new URL which includes '/commands/actions'  
    cy.url().should('include', '/commands/actions')  
  })  
})
```

Test connexion/formulaire

```
describe('Tekcollab', function() {  
  it('Error login', function() {  
    cy.visit('http://localhost:8082/tekcollab')  
  
    cy.get('#loginInput')  
      .type('ELINA')  
      .should('have.value', 'ELINA')  
  
    cy.get('#loginInput')  
      .type('password')  
  
    cy.get('input[type=submit]')  
      .click()  
  
    cy.contains('Identifiant ou mot de passe invalide')  
  })  
})
```

```
describe('My First Test', () => {  
  it('Gets, types and asserts', () => {  
    cy.visit('https://example.cypress.io')  
  
    cy.contains('type').click()  
  
    // Should be on a new URL which includes '/commands/actions'  
    cy.url().should('include', '/commands/actions')  
  
    // Get an input, type into it and verify that the value has  
    cy.get('.action-email')  
      .type('fake@email.com')  
      .should('have.value', 'fake@email.com')  
  })  
})
```

Interception de requêtes

```
describe('Tekcollab', function() {
  it('Login', function() {
    cy.server()

    cy.route('/tekcollab/formulaire/BROUILLON*', 'fx:brouillons.json')
      .as('brouillons')

    cy.route('/tekcollab/formulaire/TERMINE*', 'fx:termine.json')
      .as('termine')

    cy.visit('http://localhost:8082/tekcollab')

    cy.get('#loginInput')
      .type('ELINA')
      .should('have.value', 'ELINA')

    cy.get('#j_password')
      .type('*****')

    cy.get('input[type=submit]')
      .click()

    cy.contains('Brouillons')
  })
})
```

```
describe('The Home Page', () => {
  beforeEach(() => {
    // reset and seed the database prior to every test
    cy.exec('npm run db:reset && npm run db:seed')

    // seed a post in the DB that we control from our tests
    cy.request('POST', '/test/seed/post', {
      title: 'First Post',
      authorId: 1,
      body: '...'
    })

    // seed a user in the DB that we can control from our tests
    cy.request('POST', '/test/seed/user', { name: 'Jane' }).its('body').as('currentUser')
  })

  it('successfully loads', () => {
    // this.currentUser will now point to the response
    // body of the cy.request() that we could use
    // to log in or work with in some way

    cy.visit('/')
  })
})
```