

E2E Testing Guide - PRO PDF

Overview

This project uses **Playwright** for end-to-end testing. Playwright provides reliable, fast testing across all modern browsers.

Setup

Playwright is already installed. To install browsers, run:

```
npx playwright install
```

Running Tests

Run all tests

```
npx playwright test
```

Run tests in headed mode (see browser)

```
npx playwright test --headed
```

Run tests in UI mode (interactive)

```
npx playwright test --ui
```

Run tests in debug mode

```
npx playwright test --debug
```

Run specific test file

```
npx playwright test e2e/home.spec.ts
```

Run tests in specific browser

```
npx playwright test --project=chromium  
npx playwright test --project=firefox  
npx playwright test --project=webkit
```

View test report

```
npx playwright show-report
```

Test Structure

```
e2e/
├── home.spec.ts      # Home page tests
├── auth.spec.ts      # Authentication tests
├── tools.spec.ts     # PDF tools navigation tests
├── dashboard.spec.ts # Dashboard and logged-in features
└── theme.spec.ts     # Theme switching tests
```

Test Coverage

Home Page Tests

- Page loads correctly
- Navigation links work
- Tool cards display
- Theme toggle functionality
- Login navigation

Authentication Tests

- Login form display
- Form validation
- Successful login
- Signup form display
- Navigation between login/signup

Tool Navigation Tests

- Navigate to all tools:
- Merge PDFs
- Split PDF
- Compress PDF
- Convert PDF
- Sign PDF
- Encrypt PDF
- File upload zones visible

Dashboard Tests

- Dashboard displays after login
- User menu functionality
- Navigate to Jobs page
- Navigate to Settings
- Logout functionality

✓ Theme Tests

- Toggle to light theme
- Toggle to dark theme
- Theme persistence
- Theme toggle visible on all pages

Writing New Tests

Create a new test file in the `e2e/` directory:

```
import { test, expect } from '@playwright/test';

test.describe('Feature Name', () => {
  test('should do something', async ({ page }) => {
    await page.goto('/your-page');

    // Your test assertions
    await expect(page.locator('text=Expected Text')).toBeVisible();
  });
});
```

Configuration

The Playwright configuration is in `playwright.config.ts`:

- **Base URL:** `http://localhost:3000`
- **Browsers:** Chromium, Firefox, WebKit, Mobile Chrome, Mobile Safari
- **Screenshots:** Captured on failure
- **Traces:** Captured on first retry
- **Dev Server:** Automatically starts before tests

CI/CD Integration

In CI environments, Playwright will:

- Run tests in parallel with 1 worker
- Retry failed tests up to 2 times
- Fail the build if `test.only` is found
- Generate HTML reports

Best Practices

1. Use `data-testid` for stable selectors

```
html
<button data-testid="submit-button">Submit</button>
typescript
await page.click('[data-testid="submit-button"]');
```

2. Wait for elements properly

```
typescript
```

```
    await expect(page.locator('text=Loading')).toBeVisible();
    await expect(page.locator('text=Loading')).not.toBeVisible();
```

3. Use page.goto() with waitUntil

```
typescript
  await page.goto('/', { waitUntil: 'networkidle' });
```

4. Clean up after tests

```
typescript
  test.afterEach(async ({ page }) => {
    // Cleanup code
  });
```

5. Use fixtures for authentication

```
typescript
  test.beforeEach(async ({ page }) => {
    await login(page);
  });
```

Debugging Tips

1. See the browser while testing:

```
bash
  npx playwright test --headed --debug
```

2. Pause execution:

```
typescript
  await page.pause();
```

3. Console logs:

```
typescript
  page.on('console', msg => console.log(msg.text()));
```

4. Screenshots:

```
typescript
  await page.screenshot({ path: 'screenshot.png' });
```

5. Inspect selectors:

```
bash
  npx playwright codegen http://localhost:3000
```

Test Data

The tests use the following test account:

- **Email:** john.doe@example.com
- **Password:** password123

This account is seeded in the database for testing purposes.

Troubleshooting

Tests failing with “Timeout”

- Increase timeout in playwright.config.ts
- Check if dev server is running
- Verify selectors are correct

Browser not found

```
npx playwright install
```

Tests pass locally but fail in CI

- Check environment variables
- Verify database is seeded
- Check CI-specific configurations

Resources

- [Playwright Documentation](https://playwright.dev) (<https://playwright.dev>)
- [Best Practices](https://playwright.dev/docs/best-practices) (<https://playwright.dev/docs/best-practices>)
- [API Reference](https://playwright.dev/docs/api/class-playwright) (<https://playwright.dev/docs/api/class-playwright>)
- [Trace Viewer](https://playwright.dev/docs/trace-viewer) (<https://playwright.dev/docs/trace-viewer>)