

Playwright Installation and Test Execution

Prerequisites

1. We need to have **Node.js** installed along with an **IDE** like **VS Code**.
2. **To download and install Node.js:**
 - Go to the official website: <https://nodejs.org>
 - Install the appropriate version for your system.
 - To verify the installation, open **Command Prompt (cmd)** and type:

```
node -version
```

- When we install Node.js, **npm** (**N**ode **P**ackage **M**anager) gets installed by default.
 - To verify npm installation, type:
 - `npm -version`
 - Using **npm**, we can install required dependencies.
 - 3. **To download and install VS Code:**
 - Search for "**Download Visual Studio Code**" in any browser.
 - Click on the first link and download the installer based on your system requirements.
-

Project Setup

Creating a Project Folder

- Create a new folder in your local system.
- Open this folder in **VS Code**.

Installing Playwright

We can install Playwright in multiple ways:

1. **Using VS Code Terminal**
2. **Using Mac Terminal**
3. **Using Windows Command Prompt**
4. **Using VS Code Extension**

Installing Playwright via Terminal

- Open the **terminal** in VS Code and run the following command:

```
npm init playwright@latest
```

- Follow the interactive setup:
 1. **It will ask "OK to proceed?"** → Type `Y` and press `Enter`.
 2. **Choose TypeScript or JavaScript** → Select **JavaScript** and press `Enter`.
 3. **Where to put your end-to-end tests?** → By default, it will be placed under the `tests/` folder.
 4. **Add a GitHub Actions workflow?** → Choose **Yes (true)** or **No (false)**.
 5. **Install Playwright browsers?** → **Default is true** (recommended).

After installation, Playwright will download the required browsers and create the following files:

```
playwright.config.ts
package.json
package-lock.json
tests/
  example.spec.ts
tests-examples/
  demo-todo-app.spec.ts
```

Understanding Installed Files

Before Playwright Installation:

- Only the project folder exists.

After Playwright Installation:

- New files and folders are added to the project structure.

Project Folder Structure:

```
PlaywrightProject/
├── node_modules/           # All Playwright-related modules
├── tests/                  # Default folder where tests are created
├── tests-examples/        # Contains demo test files for reference
├── playwright.config.ts   # Manages project configurations
├── package.json            # Manages dependencies (similar to pom.xml in
Maven)
└── package-lock.json      # Tracks exact versions of installed dependencies
```

Key Configuration Files:

- **`playwright.config.ts`** → Manages **test directories, browser selection, and parallel execution**.
- **`package.json`** → Manages project dependencies and configurations.

Example playwright.config.ts:

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

export default defineConfig({
  testDir: 'tests', // Directory where test cases are stored
  fullyParallel: true, // Run all tests in parallel
  forbidOnly: !!process.env.CI, // Fail build if test.only is left
  retries: process.env.CI ? 2 : 0, // Retries on CI
  workers: process.env.CI ? 1 : undefined, // Control parallel execution
  reporter: 'html', // Generate an HTML test report

  use: {
    baseURL: 'http://127.0.0.1:3000', // Base URL for testing
    trace: 'on-first-retry', // Collect trace for failed tests
  },

  projects: [
    {
      name: 'chromium',
      use: { ...devices['Desktop Chrome'] },
    },
  ],

  webServer: {
    command: 'npm run start',
    url: 'http://127.0.0.1:3000',
    reuseExistingServer: !process.env.CI,
  },
});
```

Checking Playwright Version:

To check the installed Playwright version, open the terminal in **VS Code** and type:

```
npm playwright -v
```

Installing Playwright Using VS Code Extension

1. In **VS Code**, go to **Extensions** (left sidebar icon).
2. Search for "**Playwright Test for VS Code**" and click **Install**.
3. Open **Command Palette** (Ctrl + Shift + P or Cmd + Shift + P on Mac).
4. Search for "**Playwright**", then select "**Test: Install Playwright**".
5. A list of checkboxes will appear, select:
 - ☒ Chromium
 - ☒ Firefox
 - ☒ WebKit
 - ☒ Use JavaScript
 - ☒ Add GitHub Actions Workflow
6. Click **Install** → It will execute `npm init playwright@latest`.

Running Playwright Tests

By default, Playwright **runs in headless mode**.

To execute tests in headless mode:

```
npx playwright test
```

To run tests in headed mode (visible browser window):

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

How Playwright Executes Tests in Parallel:

- In `playwright.config.ts`, we mentioned **parallel execution** and **multiple browsers** (Chrome, Edge, and Firefox).
- If we have **2 test cases** and **3 browsers**, it will execute **6 test cases** in total.

Viewing Playwright Test Reports:

- After execution, Playwright provides a **default HTML report**.
 - To view it, run:
 - `npx playwright show-report`
 - We can also integrate external reporting tools like **Extent Reports** and **Allure Reports**.
-