**Cucumber Playwright TypeScript**

**Pre-requisites:**

**Step 1:** Download Node.js installer

* + Visit the [official Node.js](https://nodejs.org/en/download/) website to download Node.js ‘.msi’ installer.
  + Download the Windows Installer based on your system architecture (32-bit or 64-bit)

1. Run the Installer
2. Finish Setup and Install Node.js and NPM
3. Verify the Installation

Open Command Prompt or PowerShell > Check the installed versions by running these commands:

* + Type **node -v** and press Enter to check the Node.js version.
  + Type **npm -v** and press Enter to check the npm version.
  + Both commands should return version numbers, confirming successful installation.

**Step 2:** Download And Install VS Code IDE

**Framework Creation**

**Step 1:** Create an empty folder

**Step 2:** Open VS Code and go to that folder

**Step 3:** Open VS Code Terminal

**Step 4:** Execute the below command to install Playwright

**npm init playwright@latest**

**Step 5:** Delete playwright.config.ts file

**Step 6:** Delete tests-examples and tests folders

**Step 7:** Create folders src -> test -> subfolders features, and step\_definitions inside the test folder

**Step 8:** Add Cucumber plugin into VS Code IDE

Click on the Extension icon from the left panel -> Search Cucumber -> Install plugin which cucumber.io provides.

A screenshot of a computer

AI-generated content may be incorrect.

**Step 9:** Execute the command below to install Cucumber

**npm i @cucumber/cucumber -D**

**Step 10:** Execute the command below to install TypeScript node type

**npm i ts-node -D**

**Step 11:** Create cucumber.json file and add the code below

**{**

**"default": {**

**"formatOptions": {**

**"snippetInterface": "async-await"**

**},**

**"paths": [**

**"tests/features/\*.feature"**

**],**

**"publishQuite": true,**

**"dryRun": false,**

**"require": [**

**"tests/step\_definitions/\*.ts"**

**],**

**"requireModule": [**

**"ts-node/register"**

**]**

**}**

**}**

Update this file from time to time whenever it is required.

**Step 12:** In the package.json, update the script

**"scripts": {"test": "cucumber-js test"}**

Command to run tests

**npm run test**

If you are facing an import module issue, create a file named “tsconfig.json” and add the code below.

**{**

**"compilerOptions": {**

**"module": "commonjs",**

**"moduleResolution": "node"**

**}**

**}**

Hurray, you can start building a framework using Cucumber, Playwright, and TypeScript.

Taking screenshots of failed scenario and attaching it to the report

**A screen shot of a computer code

AI-generated content may be incorrect.**

Running parallel test execution, add the code below in the cucumber.json file

**"parallel": 2**

This will launch two browser instances at the same time.

**Optional:** when you want to run your automation in the Azure DevOps CI/CD. Sometimes it fails to create folders. To overcome this problem, we need to add the “fs-extra” plugin in the framework. Here are the steps to add the plugin.

**Steps:**

1. Execute this command to add fs-extra plugin

**npm i fs-extra -D**

1. Create a file called init.ts in the helper folder and add the below code in it.

**A screenshot of a computer program

AI-generated content may be incorrect.**

1. After that you need to run this file as a pretest before executing tests. Add the below line in the package.json

A screen shot of a computer

AI-generated content may be incorrect.

If you want to run your automation in multiple environments, then you need to install the below dependencies

**npm i dotenv -D**

**npm i cross-env -D**

You can add logger as well and it’s very easy to do. And it works well with parallel execution.

**npm i winston -D**

You can execute tests based on tags, in the CLI, execute the below command

**npm run test --TAGS="@reg"**

**npm run test --TAGS="@reg" --BROWSER="firefox"**

To achieve this, you will have to change cucumber.json to cucumber.js. Then only you will be able to get the value of cli.