**How to create a puppeteer project using Node.js in Visual Studio Code?**

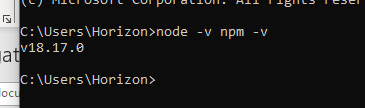
Creating a simple Puppeteer project involves setting up a basic Node.js project, installing Puppeteer, and writing a script to automate browser interactions. Below is a step-by-step guide:

Pre-requisites:

To install Node.js, you can use an installer provided on the official Node.js website. The website offers installers for various operating systems. Here are the general steps:

**For Windows:**

1. **Download:**
   * Visit the official Node.js website: <https://nodejs.org/>.
   * Click on the "LTS" tab to download the Long-Term Support version (recommended for most users).
   * Download the installer for Windows.
2. **Install:**
   * Run the downloaded installer.
   * Follow the prompts in the Node.js Installer. Accept the default settings and click "Next" through the installer.
3. **Verify Installation:**
   * Open a command prompt or PowerShell window.
   * Run the following commands to check if Node.js and npm (Node Package Manager) are installed:



* + You should see version numbers for Node.js and npm, indicating a successful installation.

**1. Initialize a Node.js Project**

Create a new directory for your Puppeteer project and initialize a new Node.js project:

A screenshot of a computer

Description automatically generated

**A screen shot of a computer

Description automatically generated**

**2. Install Puppeteer**

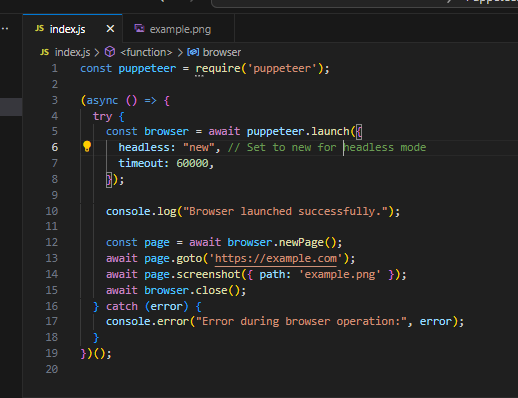
Install Puppeteer as a dependency:

**A screen shot of a computer

Description automatically generated**

**3. Create a Puppeteer Script**

Create a index.js file under your project and add your code in it.

****

1. **Run the Script**

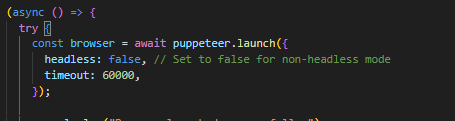
A screen shot of a computer program

Description automatically generated

This script launches a headless browser, navigates to "[https://example.com](https://example.com/)", takes a screenshot, and saves it as "example.png". You can modify the script to perform various browser automation tasks based on your requirements.

**Additional Notes:**

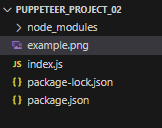
* Puppeteer will download a recent version of Chromium during installation. It's headless by default, but you can launch a non-headless browser by passing the headless: false option when launching it.



You will also observe that a screen shot of the website is taken while executiuon.

**A screenshot of a computer

Description automatically generated**

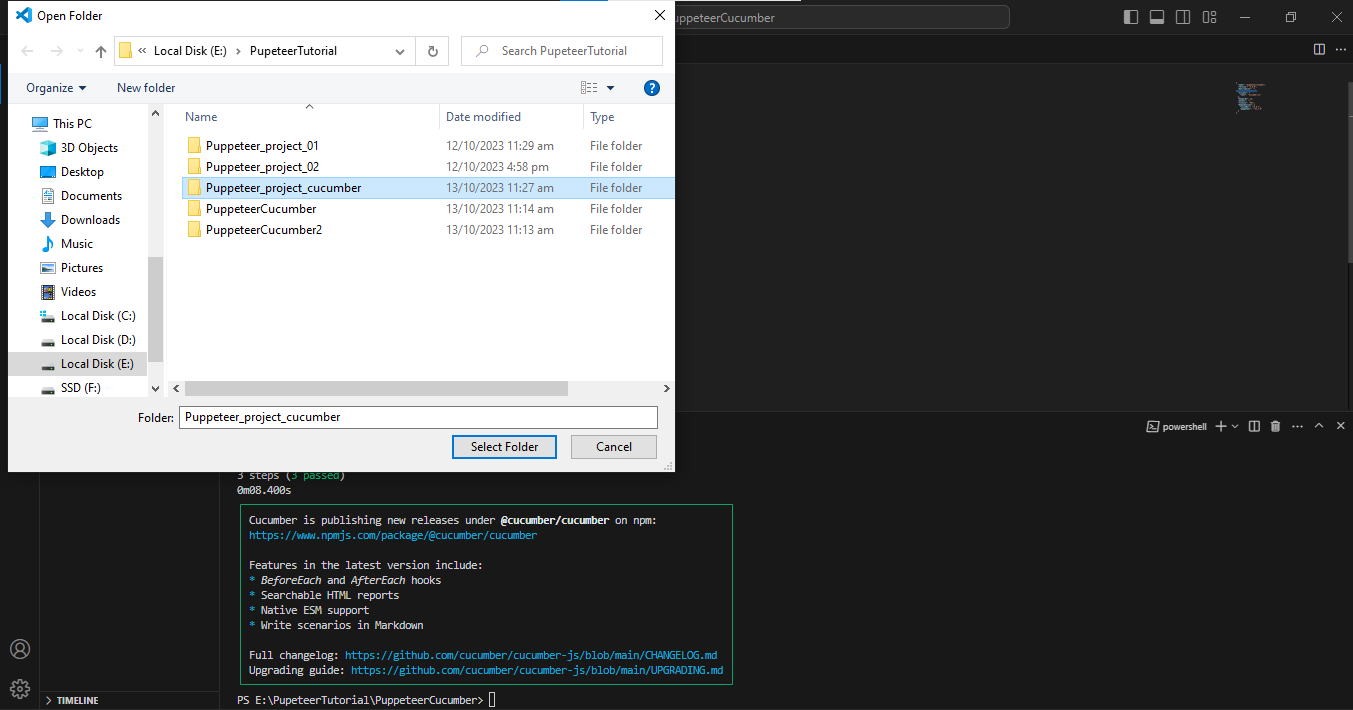
****

**Integrating puppeteer with cucumber**

Integrating Puppeteer with Cucumber involves combining the power of Puppeteer for browser automation with the behavior-driven development (BDD) approach provided by Cucumber. Below is a detailed guide on how to set up and integrate Puppeteer with Cucumber.

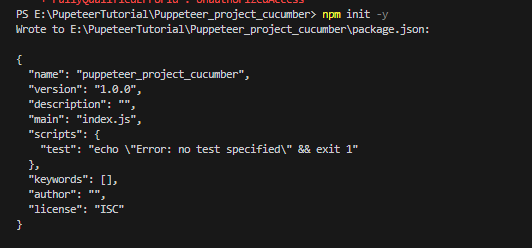
**1. Project Setup**

* Create a new directory for your project.
* I have created “Puppeteer\_project\_cucumber” folder.
* Open your folder in Visual Studio Code.



**2. Initialize a new Node.js project:**

* Open your project terminal in VS and paste this code “npm init -y”.



**3. Install Dependencies**

You need to install the dependencies by passing this code “npm install --save puppeteer cucumber” in your project terminal and all the dependencies will get installed.

* **Puppeteer:** For browser automation.
* **Cucumber:** For BDD-style testing.

A screenshot of a computer

Description automatically generated

After you have installed the dependencies, your project will look like :

A screenshot of a computer

Description automatically generated

**4. Feature File**

* Create a **features** folder under your project
* Create a feature file (**demo.feature**) under the **features** directory:
* Add your gherkin steps in demo.feature

A screenshot of a computer

Description automatically generated

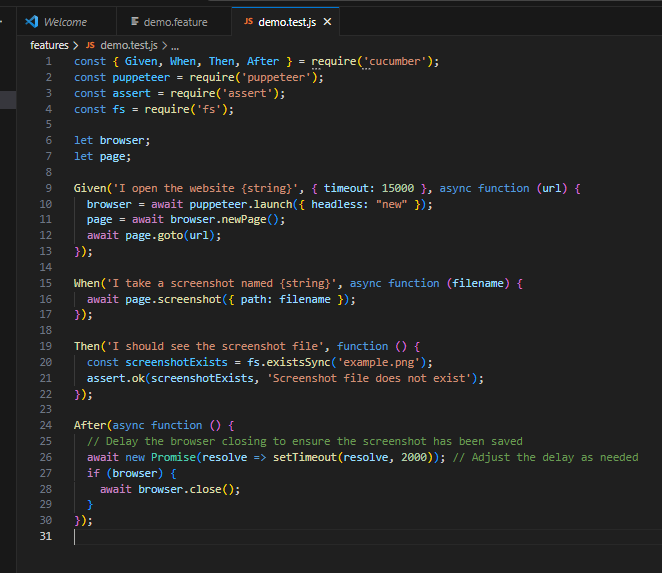
**5. Step Definitions**

* Create a step definition file (**demo.test.js**) under the samedirectory that you have placed your demo.feature:

A screenshot of a computer

Description automatically generated

* Add you steps in this newly created file.



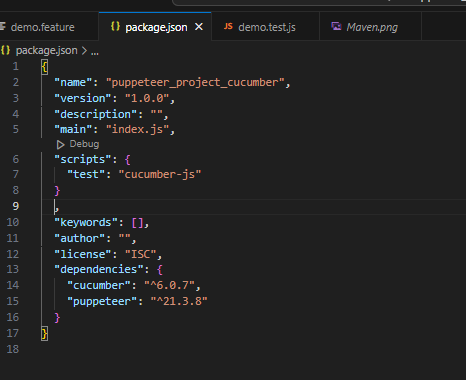
**6. Package.json Scripts**

* Update the "scripts" section in your **package.json** by adding this code:

"scripts": {

"test": "cucumber-js "

},



**7: Run your project**

* Enter “npm test” in your console and press enter.
* You will see that your test case is being executed.

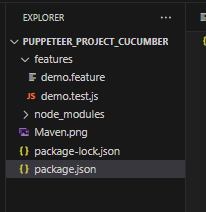
A computer screen shot of a computer program

Description automatically generated

**8: Observe the screenshot:**

As we have mentioned in our steps while execution when the website is launched then take a screenshot of the website and save the screen shot as Maven.png in the project file.

The screen shot is shown below:

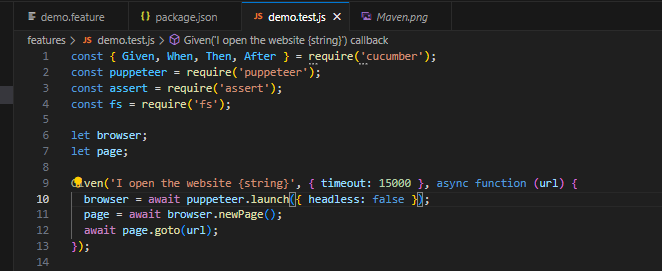


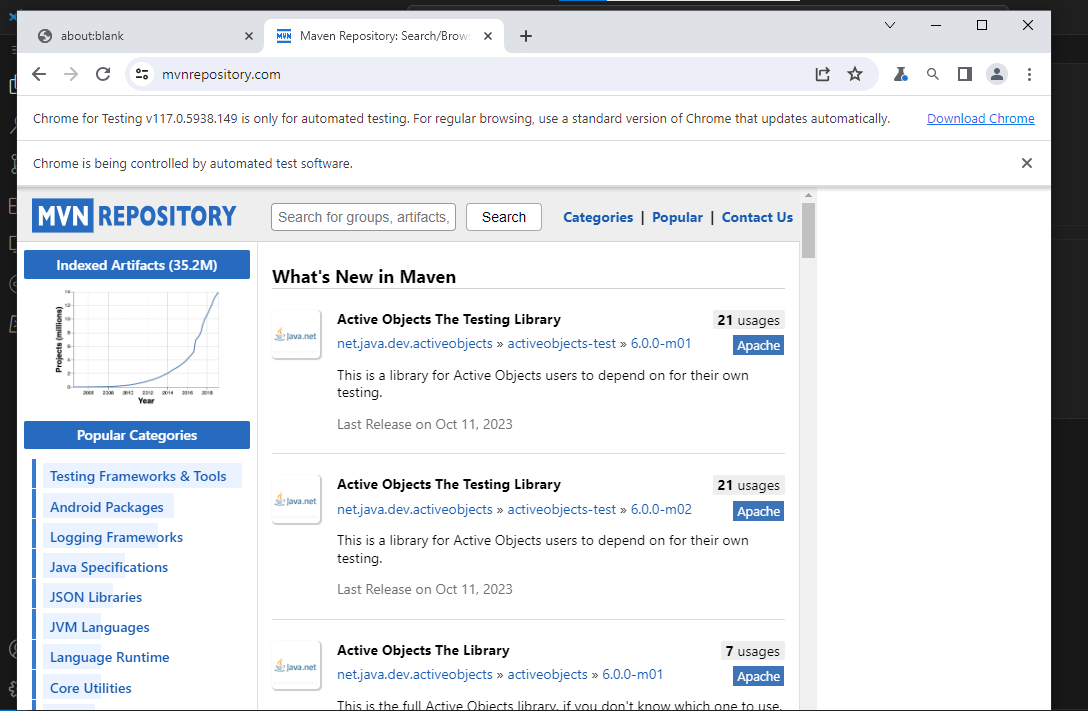
A screenshot of a computer

Description automatically generated

**9: Headful Run:**

* You can your project headful as well by modifying your launch type as “false”.





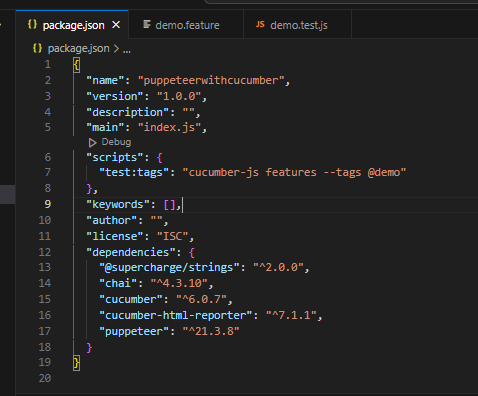
**10: Running the program with feature tags:**

* Try updating your script in **package.json** to explicitly include the **--tags** option with the desired tag:

Code

"scripts": { "test:tags": "npx cucumber-js features --tags=@demo" }

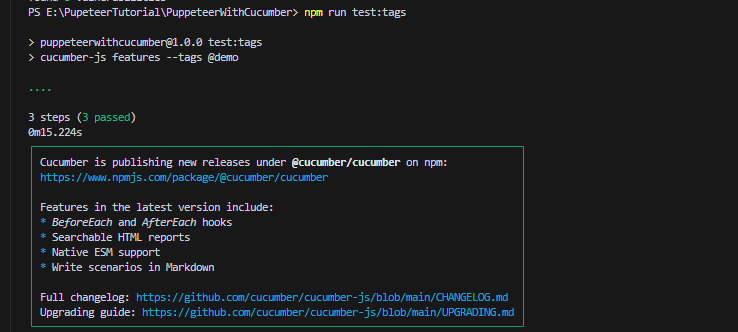
* Your package.json will look like:



* Make sure there is no space between **--tags=** and the tag value. With this modification, you should be able to run by pasting below code:

npm run test:tags

This will execute Cucumber and run only the scenarios with the **@demo** tag.



**What are the prerequisites of cloning a puppeteer project from git repository?**

To clone a Puppeteer project from a repository (e.g., a Git repository), you'll need to have a few prerequisites in place. Puppeteer is a Node.js library for controlling headless Chrome or Chromium browsers, so you should ensure that your environment is set up properly. Here are the typical prerequisites:

1. **Node.js**: Puppeteer is built on top of Node.js, so you need to have Node.js installed on your machine. You can download and install Node.js from the official website: <https://nodejs.org/>
2. **Git**: You'll need Git to clone the project from a Git repository. You can download and install Git from the official website: <https://git-scm.com/>
3. **A code editor:** You'll need a code editor to work with the Puppeteer project. Popular choices include Visual Studio Code, Sublime Text etc. You can choose the one that you are most comfortable with.

Once you have these prerequisites in place, you can proceed to clone a Puppeteer project from a repository. Here are the general steps:

1. Open a terminal or command prompt on your computer.
2. Navigate to the directory where you want to clone the Puppeteer project.
3. Use the **git clone** command to clone the repository. For example:

git clone https://github.com/username/puppeteer-project.git

Replace the URL with the actual URL of the Puppeteer project repository you want to clone.

1. After cloning the project, navigate to the project directory in your terminal using the **cd** command. For example:

cd puppeteer-project

1. Once you're inside the project directory, you can install the project's dependencies using npm (Node Package Manager) by running:

npm install

1. Now you can start working with the Puppeteer project.