# My Learning Journey – Cypress

This document describes the steps, experiences and discoveries I have made on ongoing journey to learn about the Cypress Test Automation Framework. It also covers what is needed to setup Cypress and even has some examples broken down.

# What is Cypress?

Cypress is a UI Test Framework which is Open Source

It is based on JavaScript

It is used to validate front end functionality / journeys and allows for specific checks to be made e.g. “Does Button A display? – If not, the test will fail”

Will execute tests via Chrome only

# What are the benefits?

* Easy to set-up framework
* Lots of accessible support material / training courses
* Allows for quick start test scripting even for people who have no coding background (e.g. the author of this document)
* Execution of tests are very quick due to the fact the script runs alongside the code in the Chrome browsers
* Has the capability to be integrated into Jenkins

# How did I get started?

## Mentoring

* I socialised my personal goals for 2020, which one of them was to become more aware of coding techniques and how I could apply this to my current role in Testing. This resulted in an offer of mentoring about Cypress.
* 1-2-1 sessions via Skype going through the basics
  + This is an extremely valuable way of learning, observing someone put it in context

## Online courses

### Test Automation University

* Course title : *Introduction to Cypress by Gil Tayar* <https://testautomationu.applitools.com/cypress-tutorial/>
* Highly recommended starting point - in **parallel** with mentoring.
* The course itself is 1 hour 26 minutes and is taken at a nice pace
* To get the most out of this course ***I would recommend:***
  + **Not rushing** through this course
  + Stop and Start at **your own pace.**
  + Allow for **at least double** the time of the course duration
  + ‘Play along’ as you are watching. Download the tools Gil is using (Visual Studio Code) practice different techniques that are being shown and then continue.
  + Whilst initially you may think this takes quite a lot of time, believe me, this is speeding up the overall learning process in the long run

### Cypress.io Website – getting started

* <https://docs.cypress.io/guides/overview/why-cypress.html#In-a-nutshell>
* This site can be referred to in parallel also – especially the about [**installing Cypress via NPM**](https://docs.cypress.io/guides/getting-started/installing-cypress.html#System-requirements)
* It is 100% worth working your way through the getting started section as this reinforces lessons learned during the :[*Introduction to Cypress course*](https://testautomationu.applitools.com/cypress-tutorial/) plus also introduces some new different styles so you can then put context into what you would use this for in the real world

# How to set-up Cypress

There are some pre-requisites that are needed before you can jump into Cypress – but do not worry these do not take long!

## Pre-requisites

### Ensure you have access to your own AWS Instance

Note: It is highly recommended that you set-up Cypress (and therefore the pre-requisites) on your AWS Dev instance. Download locally to your laptop may result in installation errors, proxy errors when trying to run NPM.

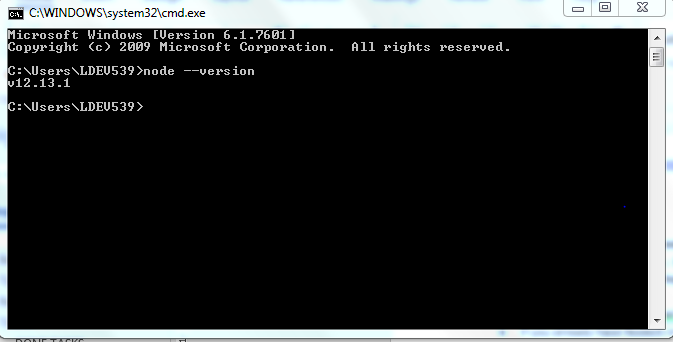
### Installing Node NPM

#### What is Node NPM?

* NPM short for Node Package Manager, is two things: first and foremost, it is an online repository for the publishing of open-source Node.js projects; second, it is a command-line utility for interacting with said repository that aids in package installation, version management, and dependency management. In this case, the open source Cypress code
* A plethora of Node.js libraries and applications are published on npm, and many more are added every day. These applications can be searched for on http://npmjs.org/. Once you have a package you want to install, it can be installed with a single command-line command.
* For more info please see : <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>

#### How to install Node NPM

* The instructions to download can be located here: <https://nodejs.org/en/>
* If you already have NodeJS installed, ensure that the version is **equal or greater than v8**. Let’s see how.
* Open a terminal window (e.g. start -run – CMD) and type: node --version



### Installing Visual Studio Code

#### What is Visual Studio Code?

* VSC is an Integrated Development Environment
* It allows you to create files, code and has a functionality called Intellisense which allows you to de-bug the content you have created.
* It also has a in-build command line / terminal so you can run the project (e.g. Cypress) from one location

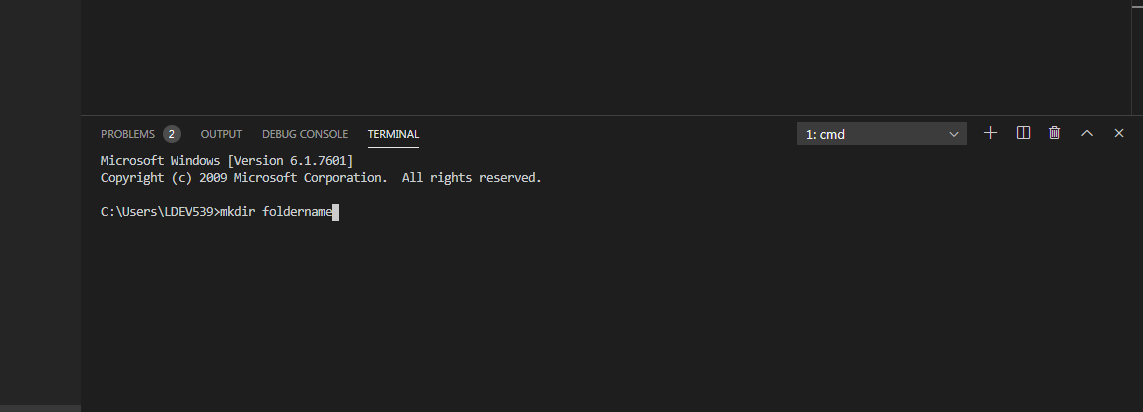
#### Installing Visual Studio Code

* <https://code.visualstudio.com/>

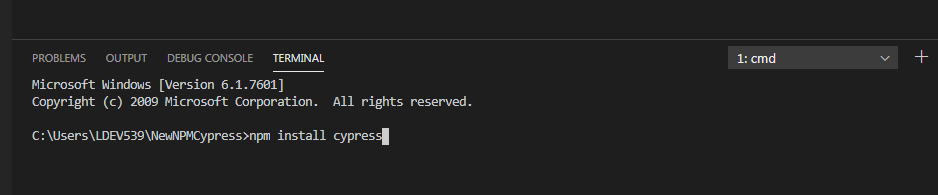
## Installing Cypress via NPM - Node Package manager

The below steps were taken from a combination of the courses/ references above in particular : <https://docs.cypress.io/guides/getting-started/installing-cypress.html#System-requirements>

1. **Create a new project folder**
   1. This is where everything to do with Cypress will be stored including files required to run
   2. Create a folder using Windows Explorer or using Visual Studio terminal window using the command : *mkdir foldername*



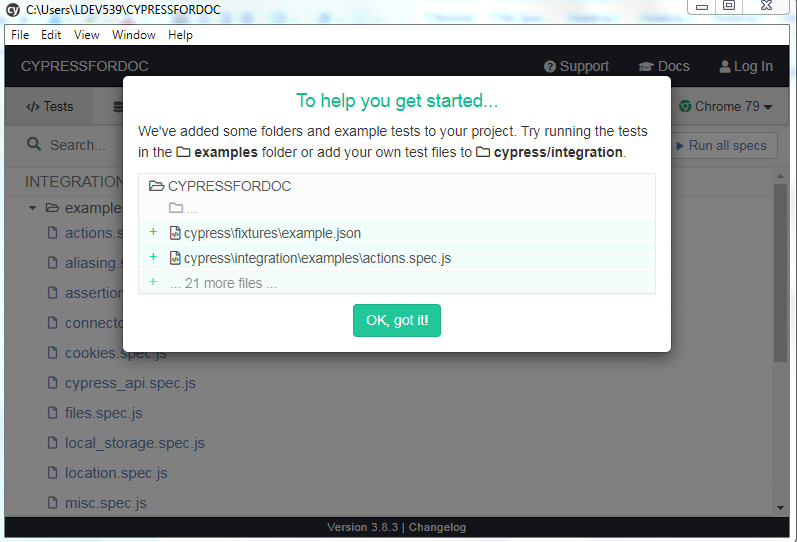
1. **Create a Package.JSON** **file in the root of the main folder created in Step 1**
   1. Open Visual Studio Code. Use File – Open Folder and navigate to the folder above
   2. Open a VSC terminal by clicking Terminal > New Terminal from the top menu bar
   3. In the terminal window which should be at the bottom of the screen, type *npm init -y.*
   4. This creates a **brand-new *package.json file*.**
      1. This is a config file for the project (i.e. the new folder you created earlier) and contains dependencies for the whole project.
      2. For example, Cypress will be a dependency for this project to run
   5. In the new project folder, there should now be a package.json file.
   6. *When we install Cypress, NPM needs to register the fact that Cypress is installed in our test folder, and it does so inside our package.json. This is why we need the package.json, and which is why we ran npm init -y.*
2. **Installing Cypress to the new folder**
   1. In the VSC terminal window, we can now download and install Cypress. including all the components it needs to run - ***to that specific folder***.
   2. On the terminal window type in: *npm install cypress*



* 1. *Note: About Cypress NPM Installation.* The first time on your computer, it might take a while, because besides the Cypress code, it will also download a separate version of Chrome that it will use to run the application under test alongside the tests.
  2. That is another thing about Cypress — it currently runs your apps only under Chrome.

## Opening Cypress

* Once all the components have installed you are now ready to open Cypress!
* Within the Visual Studio Terminal window type in: ***npx cypress open***
* The first time you open Cypress it will know this is the first time you have run Cypress and will give you a pop window indicating this.
* Now, this is the really cool thing about Cypress – **upon the first time it creates lots of examples files that you can look at in Visual Studio and examine the scripts, run in Cypress. I would highly encourage you to look at these and run them yourselves!**



# Example Cypress Test Script Files

The below describes some annotations / comments I have made for my own knowledge that help me understand break down what various commands, formats, key words mean. I thought I would share here.

***Note: These are still a work in progress and will be updated as I continue with my own journey learning Cypress***

Any items in yellow refer to the code itself, which you should be familiar with after looking at the courses

## Cypress File Structure With Narrative

/// <reference types="Cypress" />

This should go at the very top of the file. This enables intellisense for the specific file i.e. allows type ahead to give you hints and tips when scripting – very useful

Describe (also can be called Context)

This statement is used to give the **test suite a name** and wraps the individual tests case inside this statement

It (or Specify)

This statement is used to give the test case a name and belongs to the overall test suite as described above

*Cy*.

This is where the interactions which the application start to happen i.e. the test steps begin!

Although you can have commands such as Before or BeforeEach which can capture any CY commands as pre-reqs

# Structure with keywords / commands -**no syntax.**

/// <reference types="Cypress" />

Describe

It

CY.visit

# Example working file

/// <reference types="cypress" />

describe('INSERT NAME OF TEST SUITE HERE', function() {

    it('INSERT NAME OF TEST CASE 1 HERE', () => {

        cy.visit('https://example.cypress.io/')

        cy.get('body').contains('Kitchen')

      })

      it('INSERT NAME OF TEST CASE 2 HERE', () => {

        cy.visit('https://example.cypress.io/')

        cy.get('body').contains('Kitchen')

      })

})