# BDD – Behavioral Driven Development

TDD – Test Driven Development – In TDD, Developers will write dev code for features. – Testers - Exhaustive, all the Scenarios, In terms of Business.

BDD is evolved process from TDD.

BDD is a approach to Develop, test the software which aims to bridge the gap between business people and technical.

Business People: Business Analyst

Technical: Devs, Tester, Devops

There is a constant communication, feedback loops between the 3 amigos.

And There are rapid small iterations of software feature developments using the collaboration of these 3 amigos.

The focus is more on the Behavioral aspect of the product.

We are talking about the 3 Amigos – Developers, Testers, BA/ Product owners.

These 3 amigos always work together – there are always having constant communication, constant feedback is given to each other.

For every Sprint/feature/user story -> we are having the perspective of Testers(User’s point of view), Business perspective(BA), Feasibility analysis(Developer’s perspective.)

Advantages:

1. Robust Product – 3 amigos perspective
2. Constant communication
3. Constant Feedback is given
4. Speed of Development is increased
5. The product releases faster
6. Testing also involves business analysts/ Product owners.

Cucumber is a framework, BDD is not a framework. BDD is a approach.

# What is a cucumber?

Cucumber is a tool that supports Behaviour-driven development(BDD). It runs automated acceptance tests written in BDD format.

UAT – User Acceptance Testing – This is Done by Customers, Business people, End user They want to make sure that, product is ready from the End user POV(Point Of View).

It explains test steps and application behaviour using the cypress Gherkin language in simple English.

## Why use Cucumber for testing?

Cucumber is important as a testing tool for the following reasons:

* Cucumber is open-source and free to use.
* Using Cucumber, QAs can write your test scripts In multiple languages such as Java, Ruby, .NET, Python, etc.
* It integrates with selenium, Ruby on Rails, Cypress, Watir, and other web-based testing tools.
* Cucumber is one of the most widely used BDD tools.

### Cucumber Framework Components:

1. Feature files: Tests using Plain English format – Gherkin syntax (Given, When, And, Then) – Business people will write tests for User Acceptance(UAT).
2. Step Definitions: Cypress code – which translates your feature files into cypress code.
3. Runner file: this is used to run the cucumber project.(Not required for cypress).

### Feature files:

Given: Preconditions, any inputs.

When: All the user actions.

Then: Expected condition(Conclusion).

Feature: Redbus Bus Booking

Scenario: To validate booking a bus from Bangalore to Mumbai

When: User enters from city as Bangalore

And: User enters to city as Mumbai

And: User enters current date, clicks on Search button

Then: List of buses should be displayed

Feature: simple form

Scenario: To validate all the fields In simple form.

Given User has opened the browser and entered URL

When user enters firstname, lastname, email, contact and message

And user clicks on submit button

Then alert with the success message is displayed.

Feature: webtables

Scenario: To validate a text from webtable.

Given User has opened the browser and entered the URL

When user views the sortable table

Then ‘Accepted’ should be visible in 2nd row, 2nd Column.

# Installation and Configuration:

Step 1: Install Cucumber for cypress

Run the following command to install the cucumber for cypress package:

npm install –-save-dev cypress-cucumber-preprocessor

Step 2: Add the configuration cypress environment files as follows in config.js:

const cucumber = require(‘cypress-cucumber-preprocessor’) default

on(‘file:preprocessor’, cucumber())

## In the cypress.config.js file has:

const { defineConfig } = require(‘cypress’);

const cucumber = require(‘cypress-cucumber-preprocessor’).default;

async function setupNodeEvents(on, config){

on(‘file:preprocessor’, cucumber())

return config;

}

module.exports = defineConfig({

e2e: {

setupNodeEvents,

specPattern: “cypress/Integration/PageTests/\*.js”

},

});

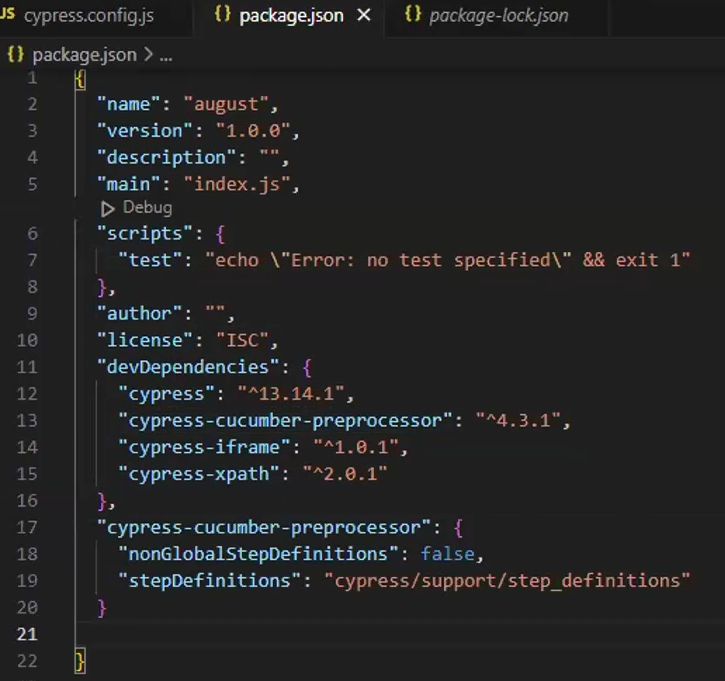
Step 3: add dependencies in package. Json as fallows:

“cypress-cucumber-preprocessor”: {

“nonGlobalStepDefinitions”: false,

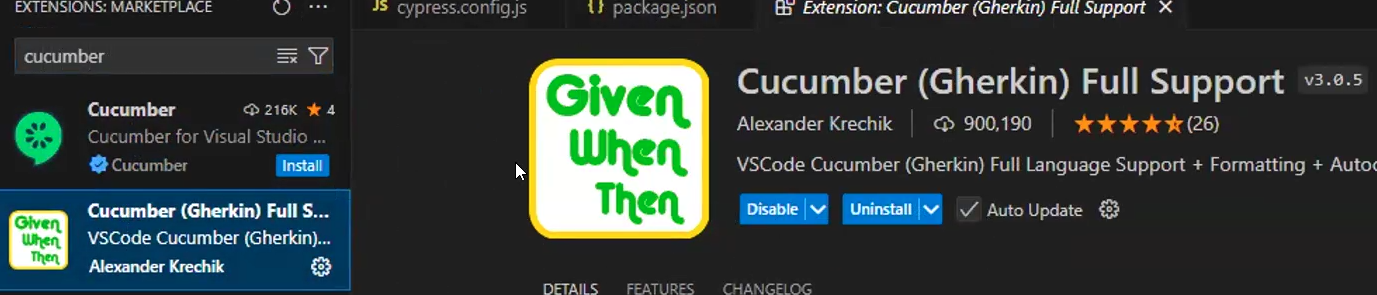
“stepDefinitions”: “cypress/support/step\_definitions”

}

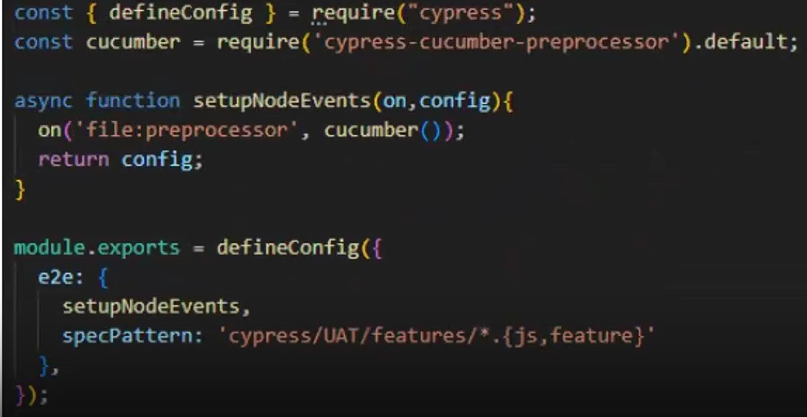


Step\_definitions is the folder present in the support folder. Instead of Step\_definitions you can provide steps as the folder

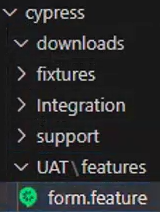
Step 4: Install



Step 5: Change the spec pattern to detect the feature files accordingly.



## Support folder > Create UAT folder > Create features folder > Create form.feature



Code in the form.feature:

Feature: Simple form

Scenario: to verify the form is working for all mandatory fields

Given open browser enter simpleform url

When User enters firstname, lastname, email, contact

And user clicks on submit

Then Successful alert should be displayed with a message

# Cypress > Support folder > steps folder > Create formsteps.js

## Code in formsteps.js:

import { Before, Given, When, Then, And } from “cypress-cucumber-preprocessor/steps”;

//Given block in step definition

Given(‘open browser enter simpleform url’, () => {

});

//When block in step definition

When(‘User enters firstname, lastname, email, contact’, () => {

});

//And block in step definition

And(‘user clicks on submit’, () => {

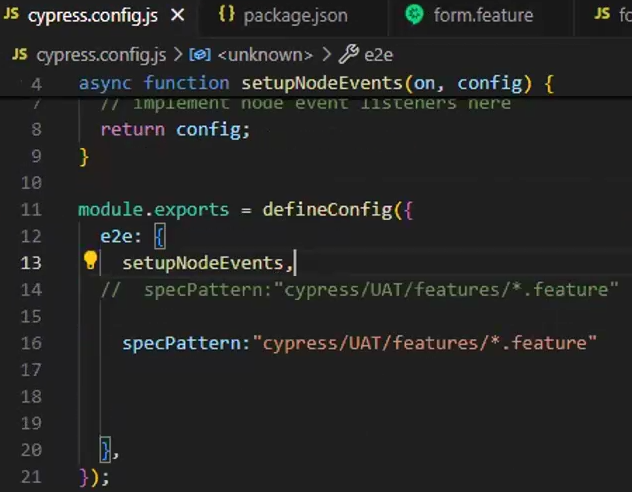
});

//Then block in step definition

Then(‘Successful alert should be displayed with a message’, () => {

});

## Copy relative path of feature files into specPattern

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Copy the relative path of the form.feature present in the UAT\features and add it to the specPattern like:

specPattern: “cypress/UAT/features/\*.feature”