

FAQs for BDD with Selenium and Specflow

Q. 1	What is Automation Testing?
Ans:	The “Automation Testing” automates the job of testing a software. In Automation Testing, a separate software is used to test the existing functional production software to be rolled out, based on the test cases identified. Automation Testing reduces the overall efforts and time required in regression testing and speeds up testing life cycle
Q. 2	When and Why Automation Testing?
Ans:	<p>Frequent regression testing</p> <p>Virtually unlimited execution of test cases is required</p> <p>Rapid feedback to developers</p> <p>Reduction in human efforts</p> <p>Test same application in multiple environment</p> <p>Finding defects missed in manual testing</p>
Q. 3	What Should Be Automated?
Ans:	<p>Good candidates</p> <ul style="list-style-type: none"> ▪ Tests executed for each build ▪ Business critical tests ▪ Tests that are difficult/tedious to perform manually <p>Bad candidates</p> <ul style="list-style-type: none"> ▪ Tests without predictable results ▪ Test that require variable input/responses from the tester ▪ Tests that perform operations in multiple environments
Q. 4	Languages Supported by Selenium
Ans:	<ul style="list-style-type: none"> ▪ Java ▪ C# ▪ Ruby ▪ Python ▪ JavaScript
Q. 5	Third Party Language Bindings
Ans:	<ul style="list-style-type: none"> ▪ Perl ▪ PHP ▪ Haskell ▪ Objective-C
Q. 6	Browsers Supported by Selenium
Ans:	<ul style="list-style-type: none"> ▪ Mozilla Firefox ▪ IE ▪ Google Chrome ▪ Opera
Q. 7	Operating Systems supported by Selenium
Ans:	<ul style="list-style-type: none"> ▪ Windows ▪ Mac ▪ Linux ▪ Unix ▪ Many more.....
Q. 8	Flavors of Selenium
Ans:	Selenium Core

FAQs for BDD with Selenium and Specflow

	Selenium RC WebDriver Selenium IDE Selenium Grid
Q. 9	Why Selenium?
Ans:	Selenium is probably the best option for automated testing of Websites today. It is becoming increasingly popular and it is the first choice of automation testers as well as organizations for automating the testing of Web-based applications.
Q. 10	What is a WebDriver?
Ans:	“Web Driver” is a Web Automation Framework which is also known as “Selenium 2”. It allows you to create and execute tests against different browsers, unlike Selenium IDE which works only with Firefox. WebDriver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API. Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded.
Q. 11	WebDriver Navigation methods?
Ans:	Navigate().GoToUrl() Navigate().Refresh() Navigate().Back() Navigate().Forward()
Q. 12	How to Locate UI Elements?
Ans:	By.Id By.ClassName By.Name By.LinkText By.PartialLinkText By.CSS By.TagName By.XPath
Q. 13	What are Locators?
Ans:	Locators are the lifeblood of the tests. Using the right locator ensures the tests are faster, more reliable or has lower maintenance over releases. Locators are used to find and match the elements of your page that it needs to interact with.
Q. 14	Locators in Selenium
Ans:	ID Name Class Name CSS Selector XPath Link Text Partial Link Text Tag Name
Q. 15	What is XPath?
Ans:	Language that describes a way to locate and process items in Extensible Markup Language (XML) documents by using an addressing syntax based on a path through the document’s logical structure or hierarchy.

FAQs for BDD with Selenium and Specflow

	XPath is used in Selenium to uniquely identify an element on a Webpage as element locator.
Q. 16	Absolute XPath
Ans:	It starts from the root element of the HTML page.the root element for every HTML Page is "html". It starts with "/" (single slash).
Q. 17	Relative XPath
Ans:	It starts with "//" (double slash).
Q. 18	Types of Wait Commands
Ans:	ImplicitWait ExplicitWait SetPageLoadTimeout SetScriptTimeout Thread.Sleep(ms)
Q. 19	Implicit Wait
Ans:	Implicit wait is used to inform WebDriver that there could be cases when some elements on the webpage will not be present instantaneously. In those cases you have to wait for some time before trying to find the element. Default wait time is 0; WebDriver will try to find the element only once and after that it will throw an exception if element is not found.
Q. 20	Explicit Wait
Ans:	Wait is explicitly applied to wait for conditions. Effective kind of wait that has to be used for each element or specified element. Explicit wait tells the WebDriver to wait till expected conditions become true, if it's not satisfied it will wait for maximum timeout period before throwing a 'NoSuchElementException'. Expected conditions is mandatory once we declare explicit wait.
Q. 21	Types of Switch commands
Ans:	ActiveElement – Switches focus to element that has focus at the moment. If there's no element with focus, this method will switch to body element of the page. Alert – Switches focus to currently active dialog box DefaultContent – Switches focus to the first frame on the page or main document if page contains iframes. Frame – Switches focus to frame by given index ParentFrame – Switches focus to parent frame of currently selected frame Window – Switches focus to the window with specified name
Q. 22	Types of Window handlers
Ans:	A set of windows that web driver can interact with is called windows handles. Each window handle is identified by random string, which is different on each test run. The different window handles supported in Selenium are: <ul style="list-style-type: none"> ▪ alert ▪ frame ▪ Window
Q. 23	Types of Alert commands
Ans:	alert.Accept(); //It accepts the alert. alert.Dismiss(); // It rejects the alert. alert.Text; //It gets text from the alert box. alert.Sendkeys("Text"); // This is used to send any text to alert box if its prompt alert.
Q. 24	What's BDD?

FAQs for BDD with Selenium and Specflow

Ans:	<p>BDD is a software development technique that defines the user behavior prior to writing test automation scripts or the functional pieces of code. Behavior-driven development should be focused on the business behaviors your code is implementing: the “why” behind the code.</p> <p>It supports a team-centric (especially cross-functional) workflow. Behavior Driven development is mostly about technical insight and business knowledge. Behavior of the user is defined by a product owner/business analyst/QA in simple English.</p> <p>These are then converted to automated scripts to run against functional code.</p>
Q. 25	What’s Cucumber?
Ans:	<p>It is a Java framework for BDD, by its support for the particular set of interactions between team members and stakeholders. It lets us define application behavior in plain meaningful English text using a simple grammar defined by a language called <i>Gherkin</i>.</p> <p>Cucumber itself is written in <i>Ruby</i>, but it can be used to “test” code written in <i>Ruby</i> or other languages including but not limited to <i>Java</i>, <i>C#</i> and <i>Python</i>. Cucumber supports writing specifications in about 30 spoken languages, making it easy to deliver better for teams outside of English-speaking territories or those working on internationally targeted software.</p>
Q. 26	What’s SpecFlow?
Ans:	<p><i>SpecFlow</i> is inspired by <i>Cucumber</i> framework in the Ruby on Rails world. <i>Cucumber</i> uses plain English in the Gherkin format to express user stories. Once the user stories and their expectations are written, the Cucumber gem is used to execute those stores.</p> <p><i>SpecFlow brings the same concept to the .NET world</i> and allows the developer to express the feature in plain English language. It also allows to write specification in human readable <i>Gherkin format</i>.</p>
Q. 27	What’s Gherkin?
Ans:	<p>Gherkin is a language, that is used to write Features, Background, Scenarios, and Steps. Gherkin helps us to write concrete requirements. Gherkin files are plain text Files and have the extension. <i>feature</i>. Gherkin syntax is simple and readable.</p>
Q. 28	Feature
Ans:	<p>Feature acts as a heading for a Scenario. Feature has a list of steps to be performed in a scenario as a whole. The Feature keyword is used to describe a software feature, and to group the related scenarios.</p> <p>A Feature has three basic elements</p> <ul style="list-style-type: none"> • The keyword – Feature. • The name of the feature, provided on the same line as the Feature keyword. • An optional description that can span multiple lines <p>Feature files are the base files to hold all the scenarios.</p>
Q. 29	What’s Background?
Ans:	<p>Backgrounds are used with the scenarios which have common functionalities. They will be executed before every scenario and perform the operation specified</p>
Q. 30	What’s Scenario?

FAQs for BDD with Selenium and Specflow

Ans:	Scenarios are just the declaration of operation in plain English Text. Scenario has many steps. Scenario holds Given-When-Then-And-But syntax. In order for Specflow to execute the test, we need to define the steps in separate file, here its .cs (C# Class) file.
Q. 31	What is a Feature file?
Ans:	The main feature of the Specflow is that it focuses on Acceptance testing. It made it easy for anyone in the team to read and write test and with this feature it brings business users in to the test process, helping teams to explore and understand requirements.
Q. 32	Styles of Mapping
Ans:	Mapping can be done in 3 styles in Specflow. <ol style="list-style-type: none">1. Regular Expression in attributes2. Method Name-Underscore3. Method Name-Pascal
Q. 33	What are tags?
Ans:	<p>Tags are markers that can be assigned to features and scenarios. Assigning a tag to a feature is equivalent to assigning the tag to all scenarios in the feature file.</p> <p>A Tag Name with a leading @ denotes tag. We can write our own definitions on what's going to happen when we tag a scenario (or feature) with a certain tag.</p> <p>If supported by the unit test framework, SpecFlow generates categories from the tags. The generated category name is the same as the tag's name, but without the leading @. You can filter and group the tests to be executed using these unit test categories.</p>