|  |  |  |
| --- | --- | --- |
| Section 1 | Click Instructions | Talking Points |
| Beacon.pngimage.png  **Click Here** | 1. Ensure PartsUnlimited.sln is open in Visual Studio. 2. In Solution Explorer, navigate to project PartsUnlimitedWebsite and open file StringContainsProductSearch.cs. 3. Walk thru the code at a high-level to help audience understand the logic. | Let's look at this method that I am working on called Search that returns products that has the names matching the LINQ query |
| Beacon.pngimage.png  **Click Here** | 1. Open the Test Explorer window. | I have written some unit tests for this Search method already, and instead of Doing a “Run All Tests” to see what is passing, I am going to start Live Unit Testing. |
| Beacon.pngimage.png  **Click Here** | 1. Click Tests -> Live Unit Testing -> Start |  |
| Beacon.pngimage.png  **Click Here** | 1. Wait till the unit test coverage glyphs appear. 2. Click on an X and then scroll down to click on a Check. | I can see Red Xs, Green Checks and Blue Dashes. Each of these glyphs tells me my test coverage and if any tests touching that line are failing - Blue Dashes indicate no unit test coverage - Red Xs indicate unit test code is failing - Green checks indicate unit test code is passing.  Looks like I have a lot of failing tests!  When I click on an X or a check, I can see which tests specifically are passing or failing. I can select a test to navigate to my test method. |
| Beacon.pngimage.png  **Click Here** | 1. In the LINQ statement, remove the ToLower() function. | Let's remove the ToLower function to see if that impacts our unit tests in any way. |
| Beacon.pngimage.png  **Click Here** | 1. Point out the failing tests. | Yes, it looks like ToLower() is an important function we're calling. Without it, our unit tests are failing. |
| Beacon.pngimage.png  **Click Here** | 1. Add ToLower() back into the LINQ query. |  |
| Beacon.pngimage.png  **Click Here** | 1. Scroll down to the Depluralize method. | Once it's fixed, notice how everything is back to passing. And we didn't need to manually run our tests. It just happens.  Let's scroll down to see failing tests around the Depluralize() method. It looks like we have an off-by-one error. |
| Beacon.pngimage.png  **Click Here** | 1. Change the Substring method to fix the off by one error: 2. query = query.Substring(0, query.Length -1) | Let's fix the code. |
| Beacon.pngimage.png  **Click Here** | 1. Observe all tests are currently passing. | As you can see, the tests are all now passing. And we've done the right thing by leaving the code in a better state than when we found it.  And with Live Unit Testing we can do so easily. |