# Understanding Builder and Editor mode

## Overview

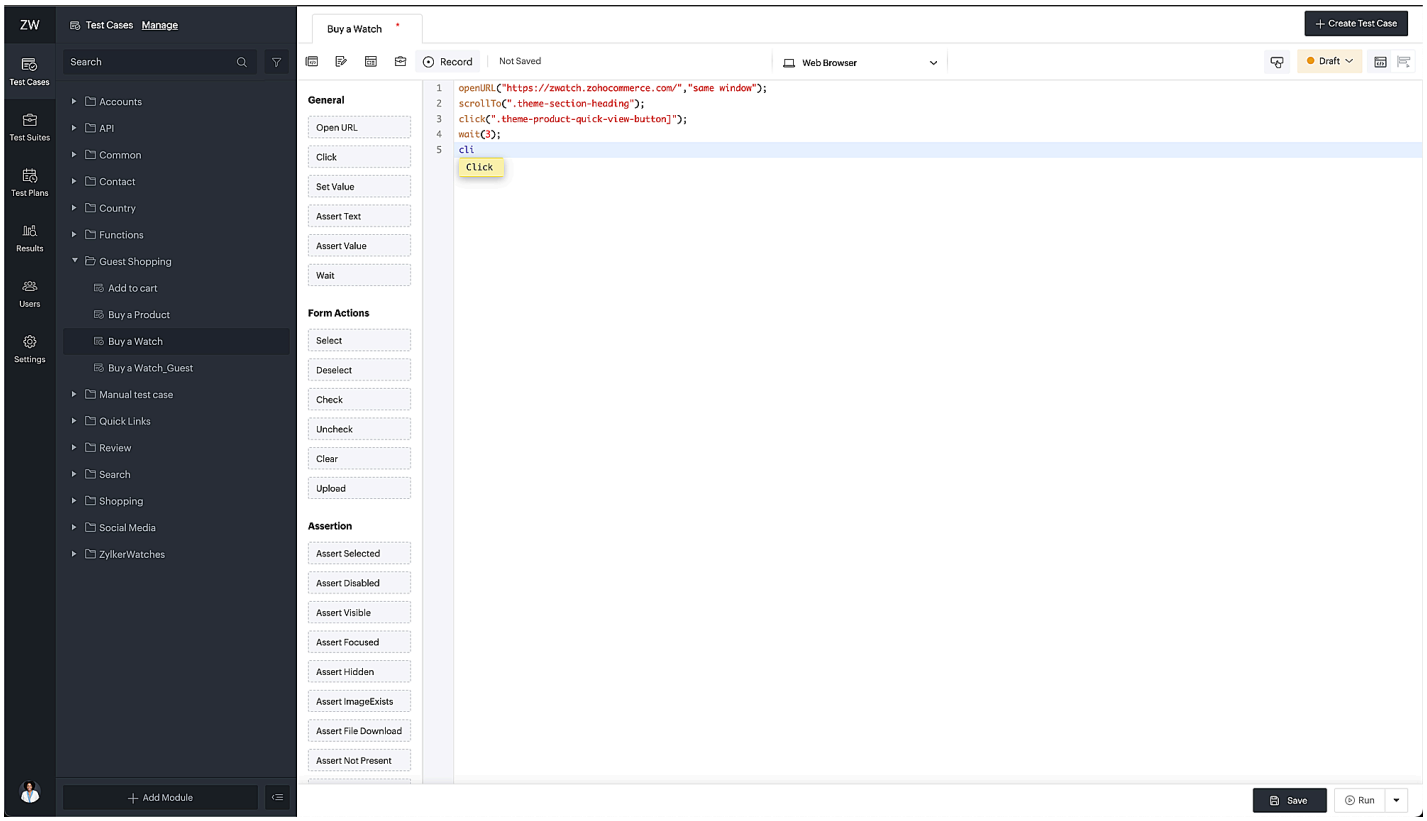
In previous help documents, we learned how to create a test case and how recording a test case can automate your testing requirements. However, to cater to different expertise and ease of use, the test editor in Zoho QEngine offers two distinct modes to write test cases: *Editor* and *Builder*. These modes are designed in such a way that the user can choose between task blocks or scripting to create their test flow.

- Editor
  The editor mode is written on a 'pro-code' basis, for those automation experts who prefers to script directly. It offers greater flexibility and control over test case scenarios, enabling users to create reusable and custom automation scripts.
- Builder
  The builder mode is the easiest one to follow, as it is created on a 'low-code' basis. Users can drag-and-drop the tasks onto the test editor, with each of them encapsulating the necessary functionality, eliminating the need for coding or writing detailed instructions manually.

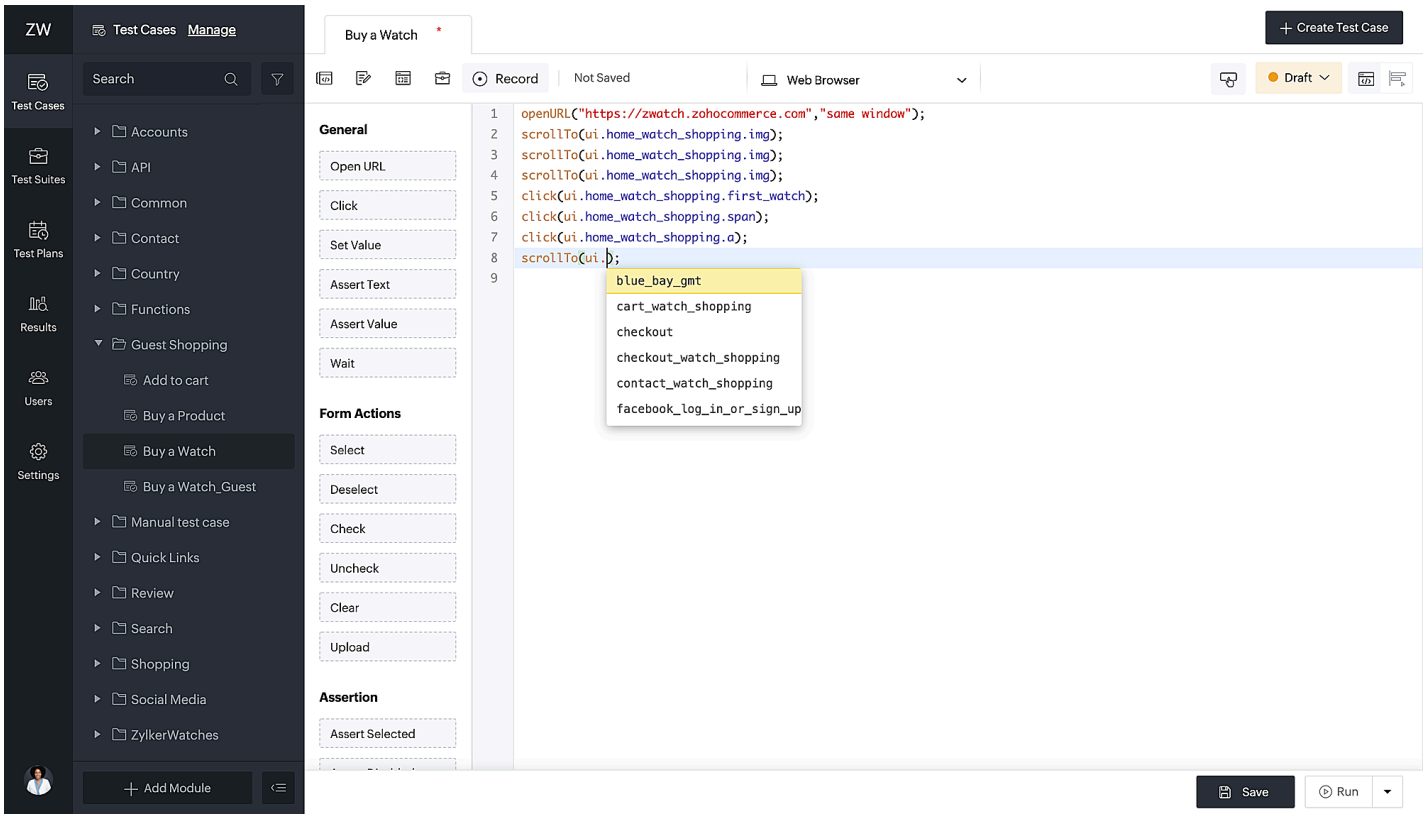
## Understanding Editor mode

Editor mode encourages a more consistent scripting style, flexible to your needs. The test editor by opens with the editor mode by default. To start writing in this mode, you can double-click or drag and drop the tasks inside the editor.

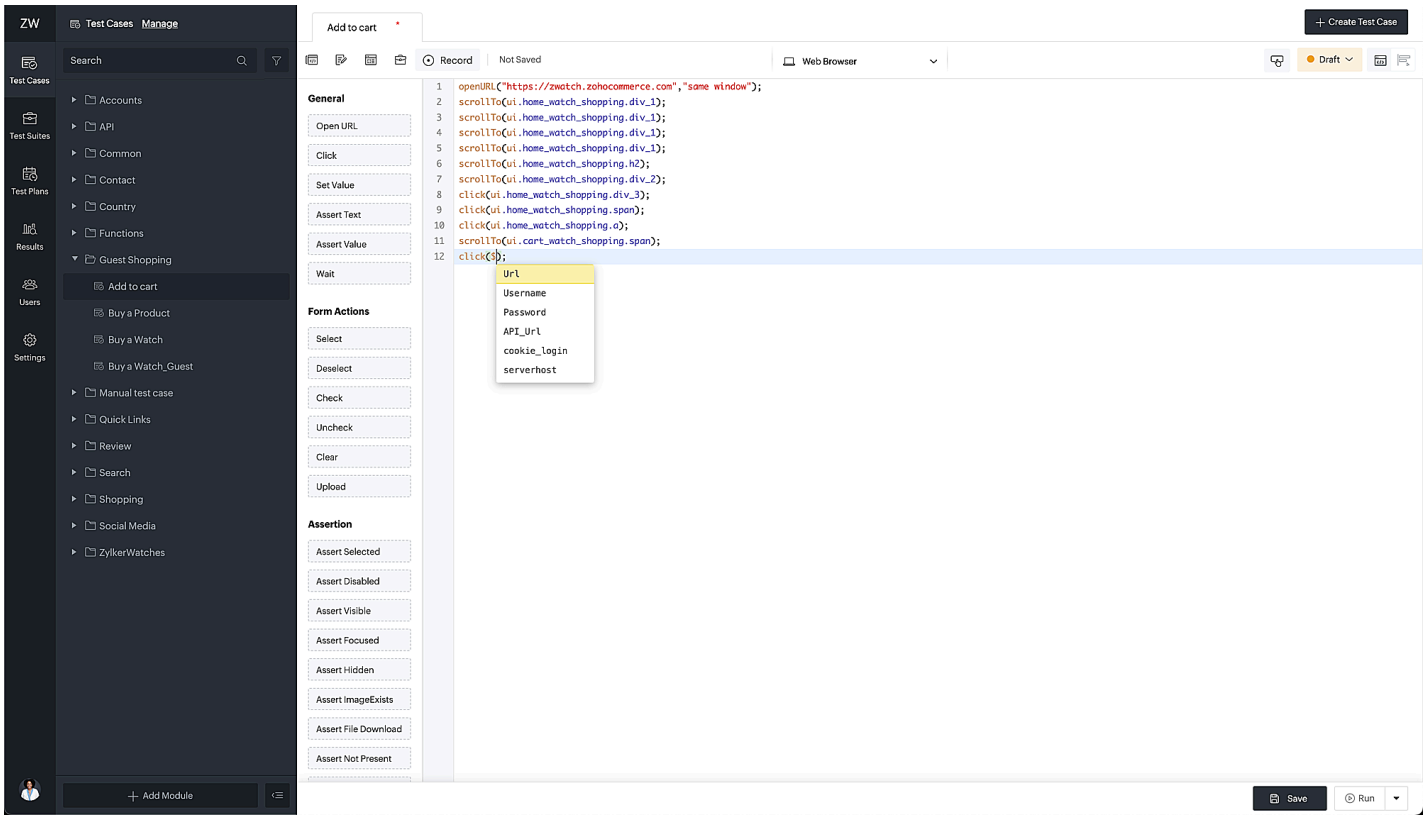
- Creating test steps becomes effortless with auto-completion suggestions for tasks, functions, and, with the right syntax, even for elements and variables. These suggestions enhance the testing process by providing quick and accurate input options.




- Type the keyword, 'ui.' to see suggestions for the saved elements. This saves you time from manually inspecting webpages to identify locators for elements every time.



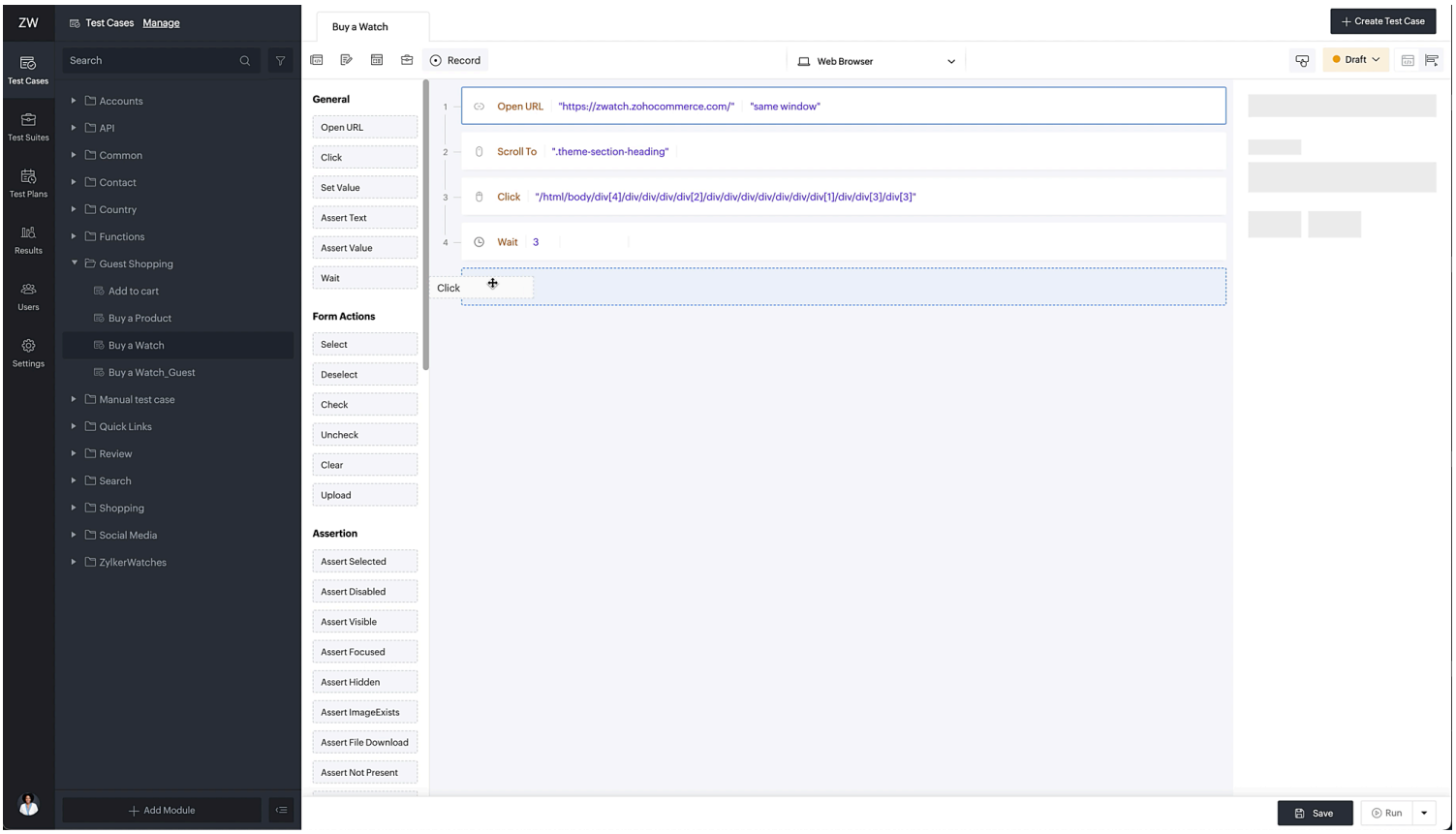

- Type '$' to see the saved variable list along with the built-in system variables.



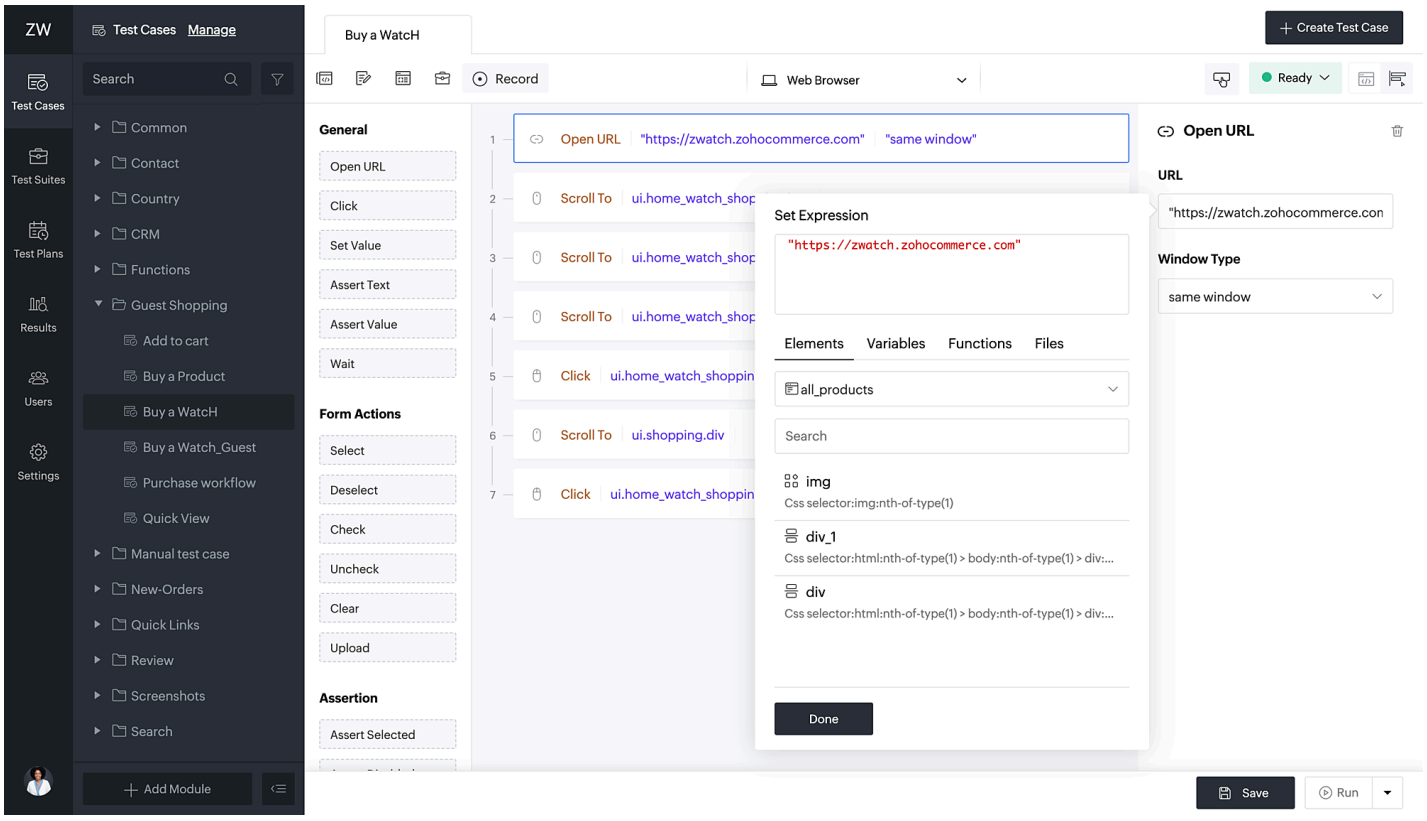

Note: Elements and variables created for a particular project can only be used for test cases inside that project.

## Understanding Builder mode

The builder mode simplifies the test case creation process. Using the same task panel, you can either drag and drop the desired tasks or double-click them to the test editor.




- The panel to the right of the test editor lists the arguments for the selected task and supplies the required values to them.




- The slider pop up helps fill in the arguments for the tasks. It provides access to the pre-saved **Elements**, **Variables**, **Functions**, and **Files**. You can either type in the arguments inside the set expression input field using

the right syntax or choose the required arguments from the four entities.

Note:

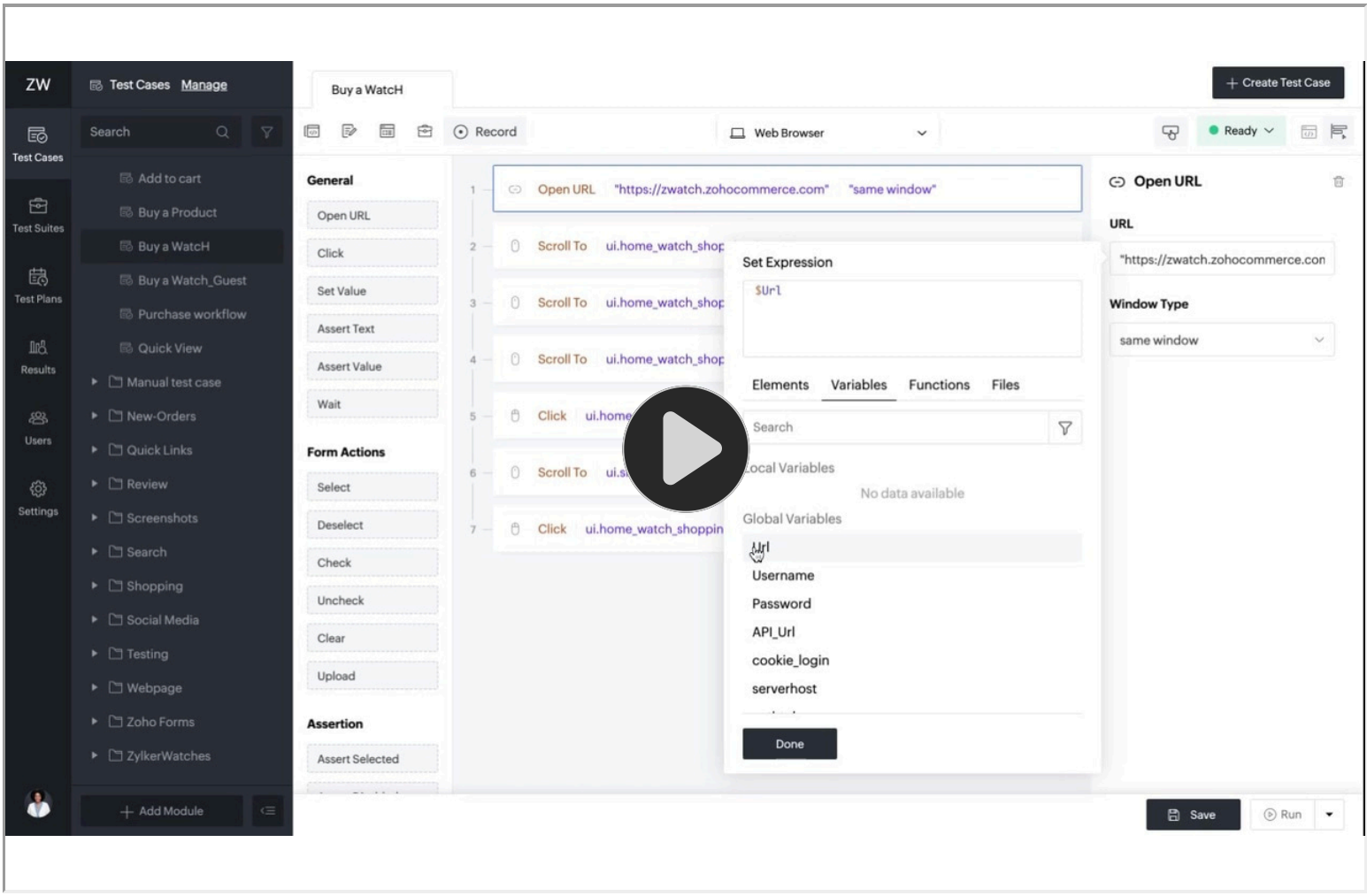
- You can reorder tasks to change the execution order. Drag and drop the required task to the required position in the editor.
- The selected task is inserted into the test editor, based on the current position of the cursor before the drag and drop or double-click actions.

## Set Expression inside Builder Mode

The builder mode argument panel is capable of accepting task parameters as it is, with the values in the format "https://www.example.com". Though it is preferred to use the four entities inside set expression pop up to avoid mistakes and for better clarity. Let's see an example and understand how the set expression pop up works using the task *Open URL*.

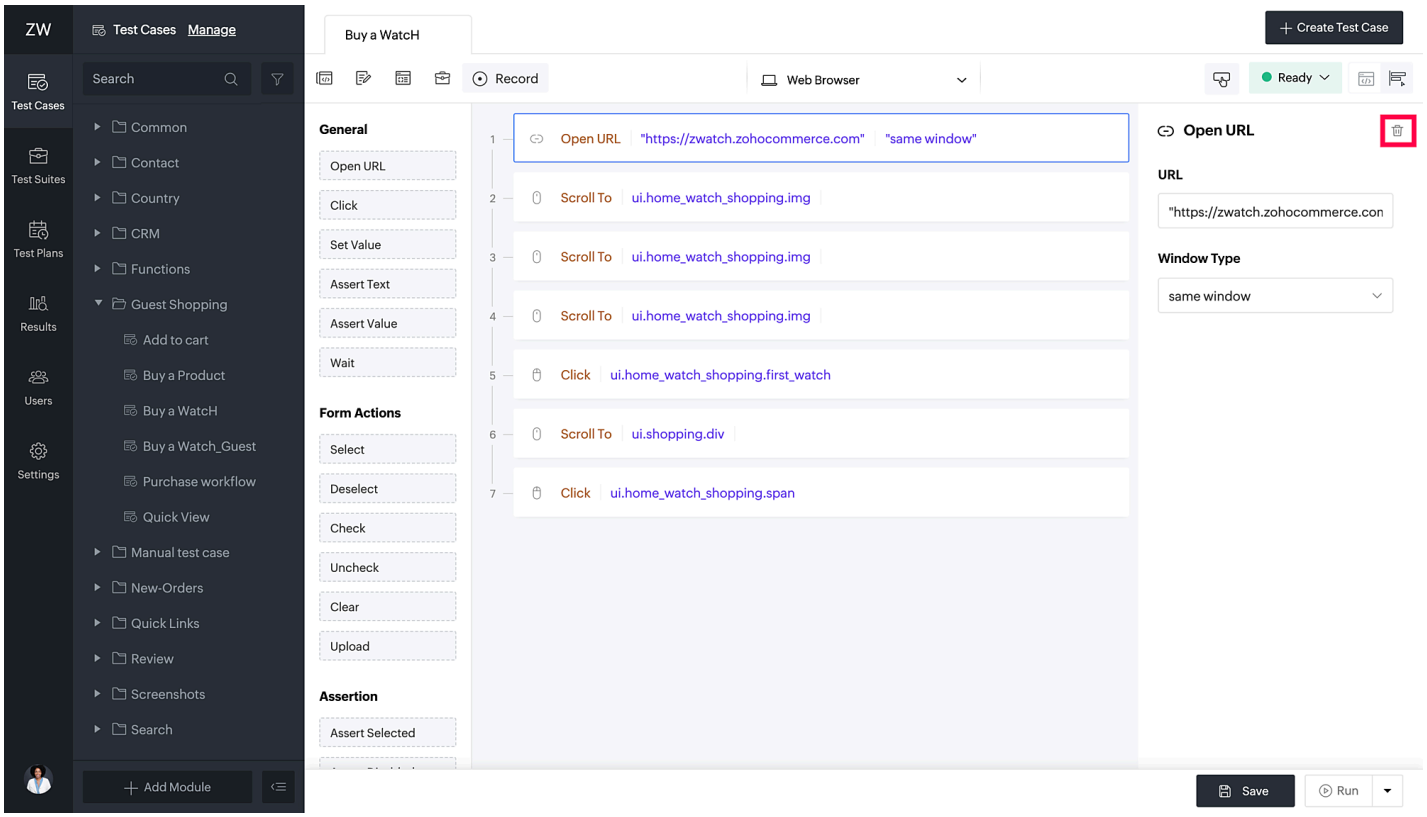
1. Drag and drop the task *Open URL* to the test editor. There are two parameters that have to be filled: *URL* and *Window Type*. As you click on the space allocated for the URL, the set expression input field gets accessed.
2. From the four entities, click **Variables**. Choose the required variable for the task from the *Global Variable* section. Click **Done**.
3. Next, click the **Window Type** and specify them using the drop-down menu: **Same Window**, **New Tab**, and **New Window**.

After providing the two parameters, the task will be automatically filled in the test editor.

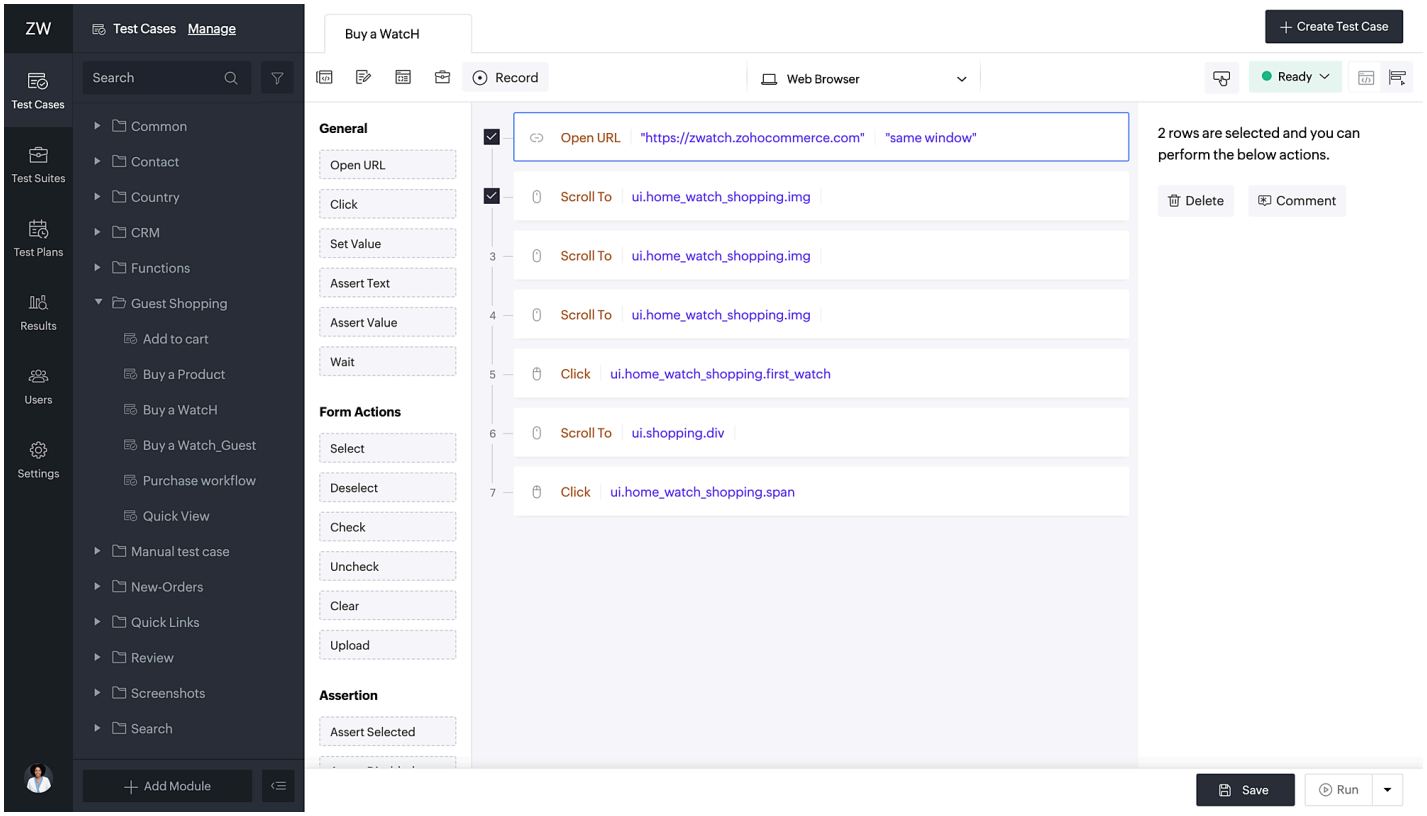



## Edit, Delete, and Comment

- To remove the task from the test case use the **trash icon**.




- To remove multiple test steps, hover over the test step you want to delete, then select the checkboxes for all the test steps you wish to remove. Click **Delete**.




- Similarly, to comment test steps, check the box you wish to comment on and click **Comment**. Do the same to uncomment on the test steps as well.

## Common actions inside Editor and Builder modes

Some common actions you find inside the test case editor for builder and editor:

- **Version** (  ) tracks and manages changes made to test cases or functions over time, ensuring version control and easy restore, if needed.
- Using the **Edit** option (  ), you can easily edit the test case details, such as test case name, link name, associated module, and the description.

Note: If the test case was marked as a function, it can't be converted back to a test case. The same applies when a test case is created.

- **Manual View** (  ) is used to create test cases that allows users to create, organize, and execute test cases manually, based on predefined steps and expected outcomes.
- **Suites** (  ) allows you to group test cases together, based on a specific testing objective or scenario.
- **Record** (  ) captures the user interactions (element locators) and actions to automate the test case creation.
- A single test case is created for testing requirements of different platforms. Use the platform switch drop-down (  ) to choose between the platforms: web, Android, iOS, and API.
- Use the refer elements (  ) option to copy or refer the existing elements. To save new locators as elements, click **+Add**. Learn more.
- The test case *status* indicates whether they are in draft or in ready state. For the test cases to be executed under a test plan, they need to be in the ready state. Use the drop-down (  ) to mark a test case or a function ready. Learn more.
- **Run** initiates live test preview for the test cases or functions to analyze and debug, if required. Customize your preferred configuration using the drop-down (  ) to execute the test cases or functions.

Note: For test cases as functions, the actions are limited. They have access to **Versions**, **Edit**, **Recorder**, **Refer Elements**, and **Switch Platform** options, as well as *run customization*.

Mobile Inspector is a common action found in both Android and iOS. It helps inspect and analyze element locators, structure, and behavior of mobile applications during testing.

By now, you should have an understanding of how to write a test case using the two modes: *Editor* and *Builder*. In the following help pages, we'll see how to create test cases for different platforms: Web, Android, iOS>and API.

# Related Links

- Web Testing using Builder and Editor mode