

Recording Clicks for Calendar Widgets

This topic provides tips on how to record clicks on calendar widgets (e.g., a travel site's calendar implemented using text input fields) in SOAtest and Virtualize.

Sections include:

- [Understanding the Default Recording Behavior for Text Input Fields](#)
- [Modifying the Recording Behavior for Text Input Fields](#)

Understanding the Default Recording Behavior for Text Input Fields

Clicks on text input fields are not recorded as a separate step by default. This is to prevent unnecessary noise. Typing username credentials into a login field, for example, would be recorded as two steps: one for clicking on the field and one for typing.

But you can change this behavior to align with the functionality you're validating. For example, travel websites often have departure and return date fields that open a calendar widget when clicked. The date is populated when the user clicks on a date element in the calendar. After creating a web recording with these actions, you may find that the click action on the calendar element was recorded, but the preceding click action on the text input field was not. The calendar element click action will fail during playback because it cannot find the calendar.

Modifying the Recording Behavior for Text Input Fields

There are three ways to modify this behavior for recording clicks on text input fields:

- Many text input fields for choosing dates have a calendar icon that opens calendar widget form. If they do, click on this icon instead of the text input field.
- Record the scenario and manually add the steps to click on the text input field later.
- Modify the UserCustomizableOptions.js file (see the following procedure for details). However, this will cause unneeded tests or actions as described above.

Modifying the UserCustomizableOptions.js Configuration File

1. Open the to the configuration file(see [Customizing Recording Options](#)).
2. Find the part of the code that resembles the following fragment:

```
ext.options.clickableInputTypes = [  
    "submit", //record click event on submit input type element  
    "button", //record click event on button input type element  
    "image", //record click event on image input type element  
    "radio", //record click event on radio input type element  
    "checkbox", //record click event on checkbox input type element "reset" //record click event on  
    reset input type element  
]
```

3. Add a text type anywhere between the brackets "[]":

```
ext.options.clickableInputTypes = [  
    "text", //text on travelocity.com  
    "submit", //record click event on submit input type element  
    "button", //record click event on button input type element  
    "image", //record click event on image input type element  
    "radio", //record click event on radio input type element  
    "checkbox", //record click event on checkbox input type element "reset" //record click event on  
    reset input type element  
]
```