

User Interface - Feature #4837

Add javascript keyboard test application for the virtual desktop

08/04/2020 08:59 AM - Sergey Ivanovskiy

Status:	WIP	Start date:	
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	50%
Category:		Estimated time:	0.00 hour
Target version:		version:	
billable:	No		
vendor_id:	GCD		
Description			
Related issues:			
Related to User Interface - Bug #3690: BUTTON widget reaction on SPACEBAR and...			Closed

History

#1 - 08/04/2020 08:59 AM - Sergey Ivanovskiy

- Related to Bug #3690: BUTTON widget reaction on SPACEBAR and ENTER keys added

#2 - 08/04/2020 09:02 AM - Sergey Ivanovskiy

This task is from [#3690-76](#) and [#3690-82](#).

#3 - 03/08/2022 08:57 AM - Sergey Ivanovskiy

- Status changed from New to WIP

Adding new test keys application dialog proved that the web application listeners continue to send events to the web client. It can be also reproduced with the existing dialogs (Log Output etc.) available from the virtual task bar menu.

p2j.screen.js exposes processMouse that can control mouse events deliveries but for the test keys it needs also to block sending key events to the web client and it needs to increase the web client inactivity timeouts watchdogTimeout and maxIdleTime so the js keys debugging doesn't break the web client. Does it make sense to add an independent web application that is opened in a tab or window having its own url like /keyboard? In this case all these issues don't affect the main web application.

#4 - 03/08/2022 09:07 AM - Greg Shah

Yes, it should not be part of the running web client, it just must be launched from the web client.

#6 - 07/08/2022 02:23 PM - Sergey Ivanovskiy

It seems that it makes sense

1. to add new functionality to record keystrokes and to load recorded key events for recorded 4GL key strokes and to replay them as a test suit on this page
2. to display the tree of performed keystrokes tests

We can use 4GL to form 4GL key strokes and then record browser key events according to this sequence of key strokes and these two files can be consider as a test case for the particular browser, keyboard and OS language. This test case can be replayed on the test page in order to verify that the keyboard driver generates the same key codes as 4GL generated key strokes.

The current test page simulates keydown, keypress and keyup events differently than the actual key events generated by the browser.

#7 - 07/08/2022 02:41 PM - Greg Shah

It is a good idea.

#8 - 07/25/2022 09:34 AM - Sergey Ivanovskiy

- % Done changed from 0 to 50

Committed rev 14105 added gui elements for test keys js application to implement replay keystrokes functionality. The rest task to implement these stub global methods

```
function shouldBeNewKeystokeTest (keyEvent)
{
    //TODO
}

function load4GLKeyCodes ()
{
    //TODO
}

function loadBrowsersRecordedKeyEvents ()
{
    //TODO
}

function addNewKeystrokesTest ()
{
    //TODO
}

function runSelectedTest ()
{
    //TODO
}
```