

Trace Run Tab

The tab consists of a toolbar and three panes:

- [Events Toolbar](#)
- [Events Pane](#)
- [Event Stack Pane](#)
- [Synchronization between the Panes and the Editor](#)
- [Quick Evaluation Pane](#)

Events Toolbar

Use the buttons on the toolbar to control the range of events to capture, configure their presentation, and navigate through the list of captured events.

Item	Tooltip and shortcut	Description
	Expand all Ctrl+NumPad+ or Ctrl+Equals	Click this button to have all the nodes in the list expanded.
	Collapse all Ctrl+NumPad - or Ctrl+Minus	Click this button to have all the nodes in the list collapsed.
	Up the Stack Trace Ctrl+Alt+Up	Click this button to navigate to the previous traced page in the stack trace.
	Down the Stack Trace Ctrl+Alt+Down	Click this button to navigate to the next traced page in the stack trace.
	Autoscroll to source	<p>Press this toggle button to have the list in the Events pane automatically synchronized with the Editor.</p> <ul style="list-style-type: none">■ When the button is pressed: as soon as you click an event in the Events pane, the details of the event are displayed in the Event Stack pane and the script that is invoked by the event is opened in the editor automatically. <p>When you navigate through the Event Stack with the Autoscroll to Trace mode turned on, the corresponding files are also automatically opened in the editor with the calling functions highlighted.</p> <ul style="list-style-type: none">■ When the button is released: the script that is invoked by the event is opened in the editor only upon double-clicking the event in the Event Stack pane.

Item	Tooltip and shortcut	Description
	Capture Events	<p>Click this button to configure the range of events to be captured and shown in the Events list.</p> <p>By default, the <i>Spy-js</i> tool captures all events on all opened Web pages, excluding <i>https secure</i> web sites, unless you have specified a URL address explicitly in the run configuration. The Events pane of the Spy-js tool window shows all captured events. If for some reasons you do not want to have all events captured, you can suppress capturing some of them by applying user-defined event filters. When you click , the pop-up list shows all the available filters, the currently applied filter is marked with a tick. By default the Capture All predefined filter is applied.</p> <p>To define a custom event filter:</p> <ol style="list-style-type: none"> 1. Click , and then choose Edit Capture Exclusions from the list. 2. In the Spy-Js Capture Exclusions Dialog that opens, click the Add toolbar button  in the left-hand pane. 3. In the right-hand pane, specify the filter name in the Exclusion name field and configure a list of exclusion rules. <p>To add a rule, click , the Add Condition to Exclusion dialog box opens. Type a pattern in the Value/pattern text box, in the Condition type drop-down list specify whether the pattern should be applied to event types or script names. Note, that glob pattern matching  is used. When you click OK, IntelliJ IDEA brings you to the Spy-Js Capture Exclusions Dialog.</p> <p>To edit a rule, select it in the list, click , and update the rule in the dialog box that opens. To remove a rule, select it in the list and click .</p> <p>To activate a filter, set a tick next to the required filter in the list.</p>
		<p>Click this button to remove all or some events from tracing and have the corresponding trace files closed in the editor. Choose one of the following options on the drop-down list that is displayed:</p> <ul style="list-style-type: none"> ■ Remove all: choose this option to cancel tracing of all captured events without closing the trace files in the editor. ■ Close all trace files: choose this option to have all trace files in the editor closed but keep tracing the corresponding event. To remove an event or script from tracing and close the corresponding trace files in the editor, choose Remove on the context menu of the event or script. ■ Remove all inactive: choose this option to remove all nodes for pages that are not active anymore (for example, because the pages have been closed in browser).

The pane shows a tree of captured events. The top-level nodes represent *documents* that are Web pages involved in tracing. When you hover the mouse over a *document*, IntelliJ IDEA displays a tooltip with the URL address of the *document*, the browser in which it is opened, and the operating system the browser is running on. The *document* node is also supplied with an icon that indicates the browser in which it is opened.

Under each *document* node, events detected on the page and scripts started from it are listed. Events of the same type are grouped into visual containers. The header of a container displays the name of the events grouped in it, the average execution time across all the events within the container, and the number of events inside the container. You can expand each node and inspect the individual events in it.

Script file names have different colour indicators to help distinguishing between them when working with the **Event Stack** pane. By hovering your mouse over a script file name, you can see the full script URL.

Once an event is clicked, its call stack is displayed in the **Event Stack** pane. The stack is represented by a tree of function calls.

Context Menu of a Document Node

Menu item	Description
Remove	Choose this option to cancel tracing all the scripts on the selected page and remove the selected node with all the items under it from the Events pane. All the currently opened trace files remain opened in the editor.
Remove all children	Choose this option to delete the items under the selected page but keep tracing it so that new events from the page are still received. The document node itself remains in the Event pane, and all the currently opened trace files remain opened in the editor.
Remove and close trace file(s)	Choose this option to cancel tracing all the scripts on the selected page, remove the selected node and all the items under it from the Events pane, and close the corresponding trace files in the editor.
Close trace file(s)	Choose this option to close all the currently opened trace files that correspond to the the selected document node and items under it. The document node and the items under it remain in the Events pane.
Refresh the page in browser	Choose this option to reload the page that corresponds to the selected document node. Tracing of the selected node is abandoned, a new document node for tracing the same page is created, and the old node becomes <i>inactive</i> .
Try closing the page in browser	Choose this option to close the page that corresponds to the selected node. Tracing of the selected node is abandoned, and the node becomes <i>inactive</i> .

Context Menu of an Event or Script

Menu item	Description
Mute event	Choose this option to add an event to an exclusion filter on the fly.
Mute script	Choose this option to add a script to an exclusion filter on the fly.

Menu item	Description
Remove	Choose this option to cancel tracing the selected event or script, remove the selected item from the Events pane, but leave the corresponding trace files opened in the editor.

Configuring the Range of Events to Capture

By default, the *Spy-js* tool captures all events on all opened Web pages, excluding *https secure* web sites, unless you have specified a URL address explicitly in the run configuration. The **Events** pane of the **Spy-js** tool window shows all captured events. If for some reasons you do not want to have all events captured, you can suppress capturing some of them by applying user-defined event filters. All the available filters are listed upon clicking the **Capture Events** button  on the toolbar, the currently applied filter is marked with a tick. By default the **Capture All** predefined filter is applied.

You can define new custom filters or add event patterns to existing filters on the fly.

- To define a new event filter:
 1. Click the **Capture Events** button , and then choose **Edit Capture Exclusions** from the list.
 2. In the **Spy-Js Capture Exclusions Dialog** that opens, click the **Add** toolbar button  in the left-hand pane.
 3. In the right-hand pane, specify the filter name in the **Exclusion name** field and configure a list of exclusion rules.

To add a rule, click , the **Add Condition to Exclusion** dialog box opens. Type a pattern in the **Value/pattern** text box, in the **Condition type** drop-down list specify whether the pattern should be applied to event types or script names. Note, that **glob pattern matching**  is used. When you click **OK**, IntelliJ IDEA brings you to the **Spy-Js Capture Exclusions Dialog**.

To edit a rule, select it in the list, click , and update the rule in the dialog box that opens. To remove a rule, select it in the list and click .

- To activate a filter, click  and set a tick next to the required filter in the list. If no filters are configured or none of the available filters fits the task, create a new filter as described above.
- While navigating through the tree of already captured events in the **Events** pane, you may come across some events or scripts that you definitely do not want to trace. You can create a filter as described above but in this case you will have to leave the pane. With IntelliJ IDEA, you can create an exclusion rule based on any event or script, as soon as you have detected such event or script, right from the **Events** pane. The rule will be either added to the currently applied filter or a new filter will be created if the current setting is **Capture All**. To add an event to an exclusion filter on the fly, select the event to exclude and choose **Mute <event name> event** or **Mute <script name> file**.

If a user-defined filter is currently applied, the new rule is added to it silently. If **Capture All** is currently active, the **Spy-Js Capture Exclusions Dialog** opens, where you can create a new filter based on the selected event or script or choose an existing filter and add the new rule to it.

Event Stack Pane

Once an event in the **Events** pane is clicked, its call stack is displayed in the **Event Stack** pane. The stack is represented by a tree of function calls. Each tree node represents the invoked function. Node text contains the total execution time, the script file name and the function name. When you click a node, the **Quick Evaluation** pane shows additional function call details, parameter values and return value, occurred exception details if there was one during the function execution.

The pane is synchronized with the editor, so you can navigate from an item in the stack tree to the corresponding *trace file* or *source file*.

- A *trace file* is a write-protected prettified version of the script selected in the **Events** pane or the script whose function is double clicked in the **Event Stack** pane. A *trace file* is named <file name>.js.trace. When you double click an item in the stack tree or select it and choose **Jump to Trace** on the context menu of the selection, the corresponding *trace file* opens in the editor with the cursor positioned at the clicked function. Another approach is to press the **Autoscroll to Trace** toggle button and select various stack nodes. In this case, the trace file opens when you click an event or script in the **Events** pane.

You can not only jump to a function but also to the place in the code where it was called from. To do that, select the required item and choose **Jump to Caller** on the context menu.

The contents of the file are highlighted to display the code execution path of the selected stack node.

- You can also navigate to the *source file* displayed as is, without prettifying, by selecting an item in the **Event Stack** pane and choosing **Jump to Source** on the context menu of the selection. If the traced site is mapped with a IntelliJ IDEA project, IntelliJ IDEA detects the corresponding local file according to the mapping and opens this file in the editor. If you are tracing a site that is not mapped to a IntelliJ IDEA project, IntelliJ IDEA opens the read-only *page source*, just as if you chose **View Page Source** in the browser.

When the traced site is mapped with a IntelliJ IDEA project, IntelliJ IDEA opens the *source file* on any attempt to edit the opened *trace file*.

Context Menu of a Script or Function

Item	Description
Jump to Caller	Choose this option to navigate to the fragment in the <i>trace file</i> from where the currently selected item was called.
Jump to Trace	Choose this option to navigate to the definition of the currently selected item in the <i>trace file</i> .
Jump to Source	Choose this option to navigate to the definition of the currently selected item in the <i>source file</i> .
Mute this File	Choose this option to add the selected script to an exclusion filter on the fly, see Configuring the Range of Events to Capture .

Synchronization between the Panes and the Editor

The **Events** and **Event Stack** panes are synchronized: when you click an event or script in the **Events** pane, its call stack is displayed in the **Event Stack** pane. To have also the corresponding trace file opened in the editor, press the **Autoscroll to Trace** toggle button on the toolbar.

The **Event Stack** pane is synchronized with the editor: when you click an item in the stack tree twice, the corresponding trace file opens in the editor with the cursor positioned at the clicked function.

To synchronize the **Events** pane directly with the editor, press the **Autoscroll to Trace** toggle button on the toolbar. In this case, as soon as you click a node in the **Events** pane, its call stack is displayed in the **Event Stack** pane and the corresponding trace file is opened in the editor. With the **Autoscroll to Trace** mode turned on, when you navigate through the **Event Stack** the corresponding files are also automatically opened in the editor with the corresponding functions highlighted.

Quick Evaluation Pane

When you click a node in the **Event Stack** pane, the **Quick Evaluation** pane shows additional function call details, parameter values and return value, occurred exception details if there was one during the function execution.

Context Menu of Function Call Details

The context menu is available from all items displayed in the pane.

Item	Description
Inspect	Choose this option to open the Inspect dialog box.
Copy Value	Choose this option to copy the value of the selected node to the clipboard.
Compare Value with Clipboard	Choose this option to open the Differences Viewer which displays the value of the selected node and the value in the clipboard so you can compare them.
Copy Name	Choose this option to copy the name of the selected node to the clipboard.

See Also

Procedures:

- [Tracing with Spy-Js](#)

Reference:

- [Spy-Js Tool Window](#)
- [Trace Proxy Server Tab](#)

External Links:

- <https://github.com/spy-js/spy-js#spy-js-documentation> 

Web Resources:

- [Developer Community](#) 