Configuring JavaScript Debugger

At the IntelliJ IDEA level, debugging of JavaScript injections is supported through the *JavaScript Debugger* bundled plugin, which is by default enabled. If not, activate it in the Plugins page of the Settings dialog box.

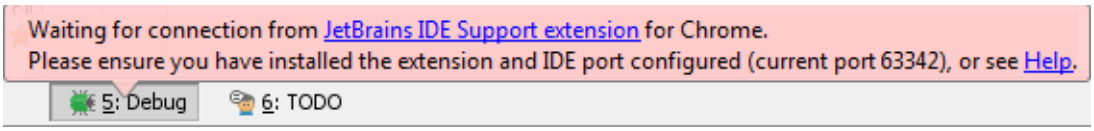
Debugging of JavaScript code is supported in the Firefox and Google Chrome browsers by means of browser-specific *JetBrains extensions*. No special steps are required from your side to install or enable these extensions. IntelliJ IDEA does it for when you initiate a debugging session for the first time.

During a debugging session, IntelliJ IDEA considers any injected JavaScript code in a Web page opened in the browser as subject for debugging. This means that you cannot use the same instance of a browser for browsing and for debugging simultaneously. For Google Chrome you can solve this problem by configuring IntelliJ IDEA to use a separate Chrome user profile for debugging.

In this section:

To configure settings required for debugging JavaScript code

1. Open the Project Settings dialog box, click **Debugger**, and select **JavaScript**.
2. On the Debugger. JavaScript page that opens, specify the following:
   - Use the **Built-in server port** spin-box to specify the number of the port where the built-in web server should run by default. During a debugging session, the *Chrome extension* will listen to this port. If for some reason the default IntelliJ IDEA port is already busy, IntelliJ IDEA finds the closest available port and starts on it. This results in a conflict: IntelliJ IDEA is running on a "new" port while the *Chrome extension* still listens to the port of a previously started product and fails to establish connection with IntelliJ IDEA. The conflict reveals when you initiate a debugging session from IntelliJ IDEA: the extension fails to connect through the default port, IntelliJ IDEA waits for connection from the extension and displays the following message with the port number where it is actually running:

     To fix the problem, specify the actual port in the Chrome extension options, see Using JetBrains Chrome Extension.
   - In the **Value tooltip delay (ms)** field, specify the time period between pointing at an object and displaying the object's value in a tooltip.
   - Whether to display DOM properties in a tab or not.
   - Whether to display function values in a tab or not. If you want the function values to be shown only for user-defined functions, select the **Show only user-defined functions**. (Otherwise, the function values will be shown for all function.)
   - Whether you want object node properties to be shown. If so, specify the properties. Use + and — to manage the list of properties.
   - Whether you want the debugger to skip certain scripts. If so, specify URLs for the corresponding scripts. Use + and — to manage the list of script URLs.

See Also

Concepts:

- Running, Debugging and Testing

Procedures:

- JavaScript-Specific Guidelines

- Debugging

Reference:

- Debugger. JavaScript

Web Resources:

- Developer Community