

## Using JetBrains Chrome Extension

The *JetBrains Chrome extension* is mainly responsible for debugging JavaScript in Chrome and Dartium. During a debugging session, the extension also supports [Live Editing](#) of HTML, CSS, and other code that contains or produces JavaScript, and provides [actual HTML DOM](#) and dynamically updated source code of the actual page.

With the *JetBrains Chrome extension*, you can run several debugging sessions controlled by several IDEs and switch between them by changing the port which the extension listens to. And finally, the extension provides the possibility to override default [CORS \(Access-Control-Allow-Origin\)](#) settings and access local scripts on your machine that are by default protected.

On this page:

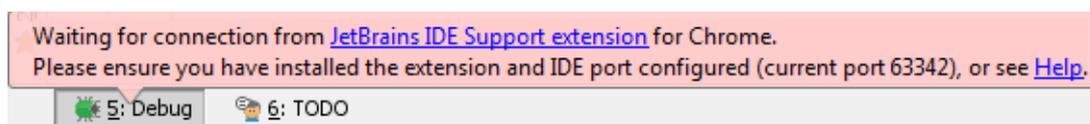
- [Installing the JetBrains Chrome extension](#)
- [Enabling, disabling, and removing the JetBrains Chrome extension](#)
- [Overriding the default CORS settings](#)
- [Changing the port for connecting to IntelliJ IDEA](#)

### Installing the JetBrains Chrome extension

You can install the extension during the first debugging session with Chrome or Dartium or at any time by visiting the *Chrome Web Store*.

1. Do one of the following:

- At any time open the *Chrome Web Store* at <https://chrome.google.com/webstore/detail/jetbrains-ide-support/hmhgeddbbohgnpmjagkdomcpobmlji>.
- *Start a JavaScript debugging session*: create a run configuration with Chrome as the debugging browser, and click the **Debug** button  on the toolbar. For details, see [Debugging JavaScript](#). IntelliJ IDEA informs you that it is waiting for connection with the *JetBrains IDE Support extension* and shows the following message in the **Debug** tool window:



Click the *JetBrains IDE Support extension* link which brings you to the *Chrome Web Store*.

In either case the **JetBrains IDE Support** page opens where you can learn more about the extension.

2. Click the **Free** button  .
3. In the **Confirm New Extension** dialog box that opens, click **Add**. The **Add to Chrome** button changes to **Added to Chrome**  .

When the extension is installed, the  icon is displayed next to the Chrome address bar.

## Enabling, disabling, and removing the JetBrains Chrome extension

Control over the JetBrains Chrome extension is provided through the `chrome://extensions/` page.

- To open the `chrome://extensions/` page, click the **Customize and control Chromium** button  next to the address bar, then choose **Tools | Extensions** on the context menu.
- To deactivate the extension, clear the **Enabled** check box. The check box name changes to **Enable**.
- To activate the extension, select the **Enable** check box.
- To uninstall the extension, click the **Remove from Chromium** button .

## Overriding the default CORS settings

Suppose the page you are debugging requests a resource which is protected against access for security reasons through CORS settings. See more about CORS at [https://developer.mozilla.org/en-US/docs/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/en-US/docs/HTTP/Access_control_CORS).

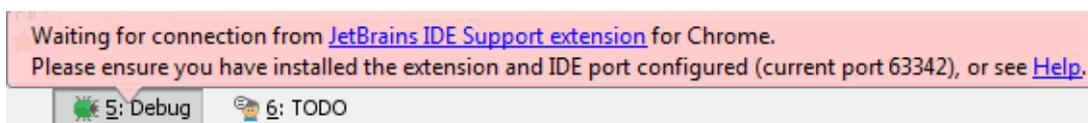
You can enable access to the resources protected by default by changing the Chrome extension options.

1. Right-click the  icon and choose **Options** on the context menu. A web page with Chrome extension options opens showing the parameters to connect to IntelliJ IDEA.
2. In the **Force CORS** text box, type the pattern that defines the URL addresses you want to make accessible, for example: `http://youtrack.jetbrains.com/rest/*`.

## Changing the port for connecting to IntelliJ IDEA

During a debugging session, the *Chrome extension* listens to the port of the software product from which the extension was invoked. Each software product including IntelliJ IDEA has its own default port on which it starts by default.

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 IntelliJ IDEA port in the Chrome extension options, see [Using JetBrains Chrome Extension](#).

1. Right-click the  icon and choose **Options** on the context menu. A web page with Chrome extension options opens showing the parameters to connect to IntelliJ IDEA.
2. In the **IDE Connection** area, specify the actual IntelliJ IDEA port in the **Port** spin box.

## See Also

Procedures:

- [Live Editing of HTML, CSS, and JavaScript](#)
- [Viewing Actual HTML DOM](#)

Reference:

- [Debug Tool Window: Elements Tab](#)

#### External Links:

- <http://blog.jetbrains.com/webstorm/2013/06/cors-control-in-jetbrains-chrome-extension/> 

#### Web Resources:

- [Developer Community](#) 