

Run/Debug Configuration: Node JS

To get access to this dialog box:

1. Make sure the *Node.js* plugin is [installed](#) and [enabled](#).

The plugin is not bundled with IntelliJ IDEA, but it is available from the [JetBrains plugin repository](#). Once enabled, the plugin is available at the IDE level, that is, you can use it in all your IntelliJ IDEA projects. See [Installing, Updating and Uninstalling Repository Plugins](#) and [Enabling and Disabling Plugins](#) for details.

2. Download and install the [NodeJS](#) framework that contains the [Node Package Manager\(npm\)](#).

In this dialog box, create configurations for running and debugging of *NodeJS* applications locally. "Locally" in the current context means that IntelliJ IDEA itself starts the *NodeJS* framework installed on your computer, whereupon initiates a running or debugging session.

This section provides descriptions of the [configuration-specific items](#), as well as the [toolbar](#) and [options](#) that are common for all run/debug configurations.

On this page:

- [Configuration Tab](#)
- [Browser / Live Edit Tab](#)
- [Toolbar](#)
- [Common Options](#)

Configuration Tab

Item	Description
Path to Node	<p>In this field, specify the NodeJS installation home. Type the path to the NodeJS executable file manually, or click the Browse button  and select the location in the dialog box, that opens.</p> <p>If you have appointed one of the installations as default, the field displays the path to its executable file.</p>

Item	Description												
Node parameters	<p>In this text box, type the NodeJS-specific command line options to be passed to the NodeJS executable file. The acceptable options are:</p> <table border="1" data-bbox="443 250 1394 1411"> <tr> <td data-bbox="443 250 1126 846"> <pre>--debug-brk=<port for connect to debugger remotely></pre> <ul style="list-style-type: none"> ■ With the <code>--debug-brk</code> option, the execution of the application suspends right after launch. This option allows you to debug the code executed on start. ■ With the <code>--debug</code> option, the code that has to be executed on the application start is executed whereupon the application waits for a debugger to connect to it. This option is useful when you are not going to debug Node.js right now, but you want to debug it later. </td> <td data-bbox="1126 250 1394 846">Specify this option to enable remote debugging of the application without re-starting the NodeJS server.</td> </tr> <tr> <td data-bbox="443 846 1126 990"> <pre>-v, --version</pre> </td> <td data-bbox="1126 846 1394 990">Print the current version of NodeJS.</td> </tr> <tr> <td data-bbox="443 990 1126 1061"> <pre>-e, --eval script</pre> </td> <td data-bbox="1126 990 1394 1061">Evaluate script.</td> </tr> <tr> <td data-bbox="443 1061 1126 1167"> <pre>--v8-options</pre> </td> <td data-bbox="1126 1061 1394 1167">Print v8 command line options.</td> </tr> <tr> <td data-bbox="443 1167 1126 1305"> <pre>--vars</pre> </td> <td data-bbox="1126 1167 1394 1305">Print various compiled-in variables.</td> </tr> <tr> <td data-bbox="443 1305 1126 1411"> <pre>--max-stack-size=val</pre> </td> <td data-bbox="1126 1305 1394 1411">Set max v8 stack size (bytes).</td> </tr> </table>	<pre>--debug-brk=<port for connect to debugger remotely></pre> <ul style="list-style-type: none"> ■ With the <code>--debug-brk</code> option, the execution of the application suspends right after launch. This option allows you to debug the code executed on start. ■ With the <code>--debug</code> option, the code that has to be executed on the application start is executed whereupon the application waits for a debugger to connect to it. This option is useful when you are not going to debug Node.js right now, but you want to debug it later. 	Specify this option to enable remote debugging of the application without re-starting the NodeJS server.	<pre>-v, --version</pre>	Print the current version of NodeJS.	<pre>-e, --eval script</pre>	Evaluate script.	<pre>--v8-options</pre>	Print v8 command line options.	<pre>--vars</pre>	Print various compiled-in variables.	<pre>--max-stack-size=val</pre>	Set max v8 stack size (bytes).
<pre>--debug-brk=<port for connect to debugger remotely></pre> <ul style="list-style-type: none"> ■ With the <code>--debug-brk</code> option, the execution of the application suspends right after launch. This option allows you to debug the code executed on start. ■ With the <code>--debug</code> option, the code that has to be executed on the application start is executed whereupon the application waits for a debugger to connect to it. This option is useful when you are not going to debug Node.js right now, but you want to debug it later. 	Specify this option to enable remote debugging of the application without re-starting the NodeJS server.												
<pre>-v, --version</pre>	Print the current version of NodeJS.												
<pre>-e, --eval script</pre>	Evaluate script.												
<pre>--v8-options</pre>	Print v8 command line options.												
<pre>--vars</pre>	Print various compiled-in variables.												
<pre>--max-stack-size=val</pre>	Set max v8 stack size (bytes).												
Path to Node App JS file	<p>In this field, specify the full path to the NodeJS file to start running or debugging from.</p> <p>If you are going to debug CoffeeScript, specify the path to the generated JavaScript file with source maps. The file can be generate4d externally or through transpilation using file watchers. For more details, see Transpiling CoffeeScript to JavaScript.</p>												
Application parameters	<p>In this text box, type the NodeJS-specific arguments to be passed to the starting NodeJS application file through the <code>process.argv</code> array.</p>												

Item	Description						
Working directory	<p>In this field, specify the working directory of the application. All references in the starting NodeJS application file, for example, <i>includes</i>, will be resolved relative to this folder, unless such references use full paths.</p> <p>By default, the field shows the <i>project root folder</i>. To change this predefined setting, choose the desired folder from the drop-down list, or type the path manually, or click the Browse button  and select the location in the dialog box, that opens.</p>						
Environment variables	<p>In this field, specify the environment variables for the NodeJS executable file, if applicable. Click the Browse button  to the right of the field and configure a list of variables in the Environment Variables dialog box, that opens:</p> <ul style="list-style-type: none"> ■ To define a new variable, click the Add toolbar button  and specify the variable name and value. ■ To discard a variable definition, select it in the list and click the Delete toolbar button . ■ Click OK, when ready <p>The definitions of variables are displayed in the Environment variables read-only field with semicolons as separators. The acceptable variables are:</p> <table border="1" data-bbox="442 1034 1394 1317"> <tbody> <tr> <td data-bbox="442 1034 746 1142">NODE_PATH</td> <td data-bbox="746 1034 1394 1142">A :-separated list of directories prefixed to the module search path.</td> </tr> <tr> <td data-bbox="442 1142 746 1249">NODE_MODULE_CONTEXTS</td> <td data-bbox="746 1142 1394 1249">Set to 1 to load modules in their own global contexts.</td> </tr> <tr> <td data-bbox="442 1249 746 1317">NODE_DISABLE_COLORS</td> <td data-bbox="746 1249 1394 1317">Set to 1 to disable colors in the REPL.</td> </tr> </tbody> </table>	NODE_PATH	A :-separated list of directories prefixed to the module search path.	NODE_MODULE_CONTEXTS	Set to 1 to load modules in their own global contexts.	NODE_DISABLE_COLORS	Set to 1 to disable colors in the REPL.
NODE_PATH	A :-separated list of directories prefixed to the module search path.						
NODE_MODULE_CONTEXTS	Set to 1 to load modules in their own global contexts.						
NODE_DISABLE_COLORS	Set to 1 to disable colors in the REPL.						
Run with CoffeeScript plugin	<p>If this check box is selected, you can launch the CoffeeScript files directly, without previously transpiling them to JavaScript.</p>						
Path to coffee executable	<p>Specify here the path to the executable file that launches the CoffeeScript transpiler.</p>						
CoffeeScript parameters	<p>Specify here the parameters to be passed to CoffeeScript.</p>						

Browser / Live Edit Tab

In this tab, enable the *Live Edit* functionality during a Node.js debugging session. This functionality is provided through a JavaScript Debug run configuration, so technically you need to create two configurations: **Node.js** for initiating a debugging session and JavaScript Debug to have *Live Edit* at disposal.

However, activating *Live Edit* in this tab lets you invoke *Live Edit* without creating a separate configuration, during a debugging session launched through the **Node.js** configuration.

For more information, see [Live Editing of HTML, CSS, and JavaScript](#).

Item	Description
After launch	<p>Select this check box to have a browser started automatically after a debugging session is launched. Specify the browser to use in the drop-down list next to the check box.</p> <ul style="list-style-type: none"> ■ To use the system default browser, choose Default. ■ To use a custom browser, choose it from the list. Note that <i>Live Edit</i> is fully supported only in Chrome. ■ To configure browsers, click the Browse button  and adjust the settings in the Web Browsers dialog box that opens. For more information, see Configuring Browsers.
With JavaScript debugger	Select this check box to enable the JavaScript debugger in the selected browser.

Toolbar

Item	Shortcut	Description
	Alt+Insert	Click this button to add new configuration to the list.
	Alt+Delete	Click this button to remove the selected configuration from the list.
	Ctrl+D	Click this button to create a copy of the selected configuration.
	Edit defaults	Click this button to edit the default configuration templates. The defaults are used for the newly created configurations.
	Alt+Up or Alt+Down	<p>Use these buttons to move the selected configuration or group of configurations (folder) up and down in the list.</p> <p>The order of configurations or folders in the list defines the order, in which configurations appear in the Run/Debug drop-down list on the main toolbar.</p>
		<p>Use this button to create a new folder.</p> <p>If one or more run/debug configurations have the focus, then the selected run/debug configurations are automatically moved to the newly created folder. If only a category has the focus, then an empty folder is created.</p> <p>Move run/debug configurations to a folder using drag-and-drop, or  buttons.</p>

Common options

Item	Description
Name	In this text box, specify the name of the current run/debug configuration. This field does not appear for the default run/debug configurations.

Item	Description
Defaults	<p>This node in the left-hand pane of the dialog box contains the default run/debug configuration settings. Select the desired configuration to change its default settings in the right-hand pane. The defaults are applied to all newly created run/debug configurations.</p>
Share	<p>Select this check box to make the run/debug configuration available to other team members.</p> <p>If the directory-based project format is used, the settings for a run/debug configuration are stored in a separate xml file in the <code>.idea\runConfigurations</code> folder if the run/debug configuration is shared and in the <code>.idea\workspace.xml</code> file otherwise.</p> <p>If the file-based format is used, the settings are stored in the <code>.ipr</code> file for shared configurations or in the <code>.iws</code> file for the ones that are not shared.</p> <p>This check box is not available when editing the run/debug configuration defaults.</p>

Item	Description		
Before launch	Specify which tasks should be carried out before starting the run/debug configuration. The specified tasks are performed in the order that they appear in the list.		
	Item	Keyboard shortcut	Description
	+	Alt+Insert	<p>Click this icon to add a task to the list. Select the task to be added:</p> <ul style="list-style-type: none"> ■ Run External tool. Select this option to run an application which is external to IntelliJ IDEA. In the dialog that opens, select the application or applications that should be run. If the necessary application is not defined in IntelliJ IDEA yet, add its definition. For more information, see Configuring Third-Party Tools and External Tools. ■ Build Artifacts. Select this option to have an artifact or artifacts built. In the dialog that opens, select the artifact or artifacts that should be built. See also, Configuring Artifacts. ■ Run Ant target. Select this option to have an Ant target run. In the dialog that opens, select the target to be run. For more information, see Ant. ■ Generate CoffeeScript Source Maps. Select this option to have the source maps for your CoffeeScript sources generated. In the dialog that opens, specify where your CoffeeScript source files are located. For more information, see CoffeeScript Support. ■ Run Maven Goal. Select this option to have a Maven goal run. In the dialog that opens, select the goal to be run. For more information, see Maven. ■ Run Remote External tool: Add a remote SSH external tool. Refer to the section Remote SSH External Tools for details.
	-	Alt+Delete	Click this icon to remove the selected task from the list.
		Enter	Click this icon to edit the selected task. Make the necessary changes in the dialog that opens.
		Alt+Up	Click this icon to move the selected task one line up in the list.
	Alt+Down	Click this icon to move the selected task one line down in the list.	

Item	Item	Keyboard shortcut	Description
	Show this page		Select this check box to have the run/debug configuration settings shown prior to actually starting the run/debug configuration.

See Also

Concepts:

- [Running, Debugging and Testing](#)

Procedures:

- [Running and Debugging Node.js](#)
- [Node.js](#)
- [Running](#)
- [Debugging](#)
- [Creating and Editing Run/Debug Configurations](#)

Reference:

- [Run/Debug Configuration: Node JS Remote Debug](#)

Web Resources:

- [Developer Community](#) 

JavaScript Support:

- [Running and Debugging JavaScript](#)