Node.js

IntelliJ IDEA supports integration with the Node.js framework thus enabling running, debugging, and unit testing of *Node.js* applications.

IntelliJ IDEA recognizes Node.js code and provides basic coding assistance and highlighting for it. To get guidance in Node development, see HowToNode.org.

In this section:

Prerequisites

1. Node.js framework is downloaded and installed on your computer.
2. 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.

Changes to the UI

The *Node.js* plugin introduces the following changes to the IntelliJ IDEA UI:

- Node.js page is added to the **Settings** dialog box.
- Run/debug configurations are added.

Enabling Node.js

1. Make sure that the Node.js framework is downloaded and installed on your computer.
2. Install and enable the *Node.js* repository plugin.
3. Restart IntelliJ IDEA for the changes to take effect.

Configuring the Node.js interpreter

1. Choose **File | Settings** and in the **Settings** dialog box that opens, click **Node.js and NPM**.
2. On the Node.js and NPM page that opens, specify the location of the desired Node.js interpreter.

   IntelliJ IDEA displays the version of the interpreter.
3. Configure the Node.js core module sources if they are not configured yet.
4. Associate the *Node.js Globals library* with the project by selecting the check box next to it on the JavaScript Libraries page of the Settings / Preferences Dialog.

Configuring Node core module sources

When developing a Node.js application it can be convenient to have code completion, reference resolution, validation, and debugging capabilities for Node core modules (`fs\`, `path`, `http`, etc.). However, these modules are compiled into the Node.js binary. IntelliJ IDEA provides the ability to configure these sources as a JavaScript library and associate it with your project.

1. Choose **File | Settings** and in the **Settings** dialog box that opens, click **Node.js and NPM**.
2. On the Node.js and NPM page that opens, click the **Configure** button. The Setting Up Node.js Sources dialog box opens.
3. In the **Node.js sources location** area, specify where to take the Node.js core module sources to use.
   - If you have not downloaded the sources in any format yet, choose the **Download from the Internet** option. IntelliJ IDEA will download them from the default location depending on the Node.js version used, arrange them in a `Node.js 0.6.12 Core Library` JavaScript library, and store the library in the IntelliJ IDEA system local folder.
   - If you have already downloaded the sources but have not extracted them from the archive, choose the **Archive file:** option. Specify the location of the downloaded archive in the **File** text box. Type the path to the file manually or click the **Browse** button … and choose the file in the dialog box that opens.
   - If you have already downloaded and extracted the sources, choose the **Directory:** option. In the **Source root directory** text box, specify the folder where the sources have been extracted to. Type the path to the folder manually or click the **Browse** button … and choose the folder in the dialog box that opens.
4. In the **Usage scope** area, specify the visibility (*Global* or *Project*) and availability of the library to be configured.
   - To restrict the visibility of the sources and the ability to reference them to the current project, select the **Associate with project** check box. IntelliJ IDEA will configure the sources as a *Project* library and automatically enable referencing them from the entire current project. However you will be unable to associate them with any other projects later.
   - To enable the use of the configured sources at the IntelliJ IDEA level, clear the **Associate with project** check box. IntelliJ IDEA will configure the sources as a *Global* library, so you will be able to associate them with other projects. However, the configured library will not be automatically associated with the current project, so you will have to configure its usage scope manually.
5. Click the **Configure** button to start configuring the sources as a JavaScript library.
6. After the library configuration is completed, customize the library usage scope, if necessary. Click the **Edit usage scope** link. In the Usage Scope dialog box that opens, click the desired directories, and from the drop-down list select the newly configured Node.js core module sources library.

   > The use of a library is enabled recursively, that is, if a library is associated with a folder it is automatically enabled in all the nested directories and files.

See Also

Procedures:

- JavaScript-Specific Guidelines
- Configuring JavaScript Libraries
- CoffeeScript Support
- Generating a Project from a Framework

Reference:

- Plugins
- Node.js and NPM
- Run/Debug Configuration: Node JS
- Run/Debug Configuration: Node JS Remote Debug
- Run/Debug Configuration: NodeUnit

External Links:

- http://blog.jetbrains.com/webide/2012/03/attaching-the-sources-of-node-js-core-modules/
- http://nodejs.org
- http://howtonode.org/

Web Resources:

- Developer Community