

Running Nodeunit Tests

IntelliJ IDEA supports integration with the *nodeunit* framework thus enabling running unit test for Node.js applications.

To create and run unit tests for a Node.js application, perform these general steps

1. [Enable nodeunit support](#).
2. [Write the unit tests](#).
3. Mark the folder where the unit tests are stored as a test source folder, see [Configuring Content Roots](#).
4. [Create a run configuration](#) of the type *Nodeunit*.
5. [Launch unit tests](#) and [monitor test results](#) in the **Run** tool window.

This topic provides guidelines in Node.js-specific unit testing procedures. For general information on testing in IntelliJ IDEA, see the section [Testing](#).

To enable unit testing for Node.js applications

1. [Install](#) and [enable](#) the *Node.js* plugin.

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 [Node.js](#) framework.
3. Download and install the [nodeunit](#) testing framework

To create Nodeunit tests, perform these general steps

- [Create a folder](#) `test` at the same level as the `src` folder
- Populate the `test` folder. For each production file, create a separate test file.
- Mark the folder where the unit tests are stored as a test source folder, see [Configuring Content Roots](#).

To create a Nodeunit run configuration

1. Open the [Run/Debug Configuration](#) dialog box by doing one of the following:
 - On the main menu, choose **Run | Edit Configurations**.
 - Press `RunConfiguration` and choose **Edit Configuration** on the context menu.
2. Click the **Add** button **+** on the toolbar and select the **Nodeunit** configuration type.
3. In the dialog box that opens, specify the following:
 1. The name to identify the configuration.
 2. The path to the NodeJS installation to use.

If you have [appointed one of the installations as default](#), the field displays the path to its executable file.
 3. The working directory. This can be the project root folder or the parent directory for the test folder.
 4. The scope of tests to run.
 - To have IntelliJ IDEA run all the test files in a folder, choose **All JavaScript test files in the directory** from the **Run** drop-down list. In the **Directory** text box, type the path to the test folder relative to the [working directory](#).
 - To have a specific test executed, choose **JavaScript test file** from the **Run** drop-down list. In the **JavaScript test file** text box, type the path to the file relative to the [working directory](#).
4. Apply the changes and close the dialog box.

See Also

Concepts:

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

Procedures:

- [Unit Testing JavaScript](#)

Reference:

- [Run/Debug Configuration: NodeUnit](#)

Web Resources:

- [Developer Community](#) 