Using JavaScript Code Quality Tools

IntelliJ IDEA provides facilities to run JavaScript-specific code quality inspections through integration with the following code verification tools:

- JSLint. The tool is bundled with IntelliJ IDEA.
- JSHint. IntelliJ IDEA comes bundled with the default version of the tool, but you can have an alternative version downloaded.
- Closure Linter. The tool is not bundled with IntelliJ IDEA, so you need to download it yourself.

All these tools check JavaScript code for most common mistakes and discrepancies without running the application.

On this page:

- Enabling and configuring the built-in JSLint tool
- Enabling and configuring JSHint
- Enabling and configuring Closure Linter

Enabling and configuring the built-in JSLint tool

1. Open the project settings, click **JavaScript**, and then under the Code Quality Tools node, click **JSLint**. The JSLint page opens.
2. Select the **Enable** check box. After that all the controls in the page become available.
3. Define the set of common mistakes to check the code for. To enable a validation, select the check box next to it.

Enabling and configuring JSHint

1. Open the project settings, click **JavaScript**, and then under the Code Quality Tools node, click **JSHint**. The JSHint page opens.
2. Select the **Enable** check box. After that all the controls in the page become available.
3. From the **Version** drop-down list, choose the version of the tool to apply.

   IntelliJ IDEA comes bundled with *version 1.0.0*, which is used by default. IntelliJ IDEA provides the ability to download another version, which is not bundled. Actually, the alternative version is downloaded only once, whereupon it is available without download.

4. Configure the behaviour of JSHint:
   - To have the code verified according to the rules from a previously created configuration file, select the **Use config files** check box.

     A configuration file is a JSON file with the extension `.jshintrc` that specifies which JSHint options should be enabled or disabled. IntelliJ IDEA will look for a `.jshintrc` file in the working directory. If the search fails, IntelliJ IDEA will search in the parent folder, then again in the parent folder. The process is repeated until IntelliJ IDEA finds a `.jshintrc` or reaches the project root. To have IntelliJ IDEA still run verification in this case, specify the default configuration file to use.

     Starting with IntelliJ IDEA version 6.0.1, the *globals* setting in `.jshintrc` is supported. For earlier versions, a *Predefined (, separated)*"tree node is displayed in the bottom of the tree in the UI.

   - To configure verification manually, clear the check box and define the set of common mistakes to check the code for in the **Options** area. The controls in the area fall into two groups:
     - **Enforcing options:** select the check boxes in this group to enable very strict behaviour of the verification tool and thus allow only safe JavaScript.
     - **Relaxing options:** select/clear the check boxes in this area to suppress warnings when certain types of discrepancies are detected.
     - **Environments:** select/clear these check boxes to specify for which environments you want global properties predefined.

     To enable a validation, select the check box next to it.

Enabling and configuring Closure Linter

1. Download and install the Closure Linter tool.
2. Open the project settings, click **JavaScript**, and then under the Code Quality Tools node, click **Closure Linter**. The Closure Linter page opens.
3. Select the **Enable** check box. After that all the controls in the page become available.
4. Select the **Enable** check box. After that all the controls in the page become available.
5. Specify the path to the `<Python_home>\Scripts\gslint.exe` file and the previously created configuration file.

See Also

Procedures:

- Code Inspection
- JavaScript-Specific Guidelines

Reference:

- JSHint
- JSLint

- Closure Linter
- Inspections

Web Resources:

- Developer Community