Run/Debug Configuration: DartUnit

To get access to this dialog box:

1. Download the Dart SDK.

2. Make sure the *Node.js* plugin is installed and enabled.

   The plugin is <u>not</u> 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.

3. Make sure the *Dart* plugin is installed and enabled.

   The plugin is <u>not</u> 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.

4. Configure the Dart SDK in IntelliJ IDEA.

In this dialog box, create configurations to run unit tests for *Dart* applications.

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

Item	Description
Test Kind	From this drop-down list, choose the scope of tests to execute. The available options are: ■ **All:** choose this option to to have IntelliJ IDEA run all the test methods in the file specified in the **Test File** text box. ■ **Method:** choose this option to to have IntelliJ IDEA run a single test method. Specify the method in the **Test Method** text box. ■ **Group:** choose this option to to have IntelliJ IDEA run tests from a group. Specify the group name in the **Test Group** text box. See more about grouping tests at https://www.dartlang.org/articles/dart-unit-tests/#grouping-tests.
Test File	In this text box, specify the file with the test methods to execute.
Method Name	In this text box, specify the test method to execute. The text box is available only when **Method** is chosen in the **Test Kind** drop-down list.
Group Name	In this text box, specify the group of test methods to execute. The text box is available only when **Group** is chosen in the **Test Kind** drop-down list.

Item	Description
VM Options	In this text box, specify the options to launch the Dart Virtual Machine with. When specifying the options, follow these rules: ■ Use spaces to separate individual options, for example, `-client -ea -Xmx1024m`. ■ If an option includes spaces, enclose the spaces or the argument that contains the spaces in double quotes, for example, `some" "arg` or `"some arg"`. ■ If an option includes double quotes (e.g. as part of the argument), escape the double quotes by means of the backslashes, for example, `-Dmy.prop=\"quoted_value\"`.
Program arguments	In this text box, type the list of arguments to start the tests with. Use the same format as for **VM options**.

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.
⬆ or ⬇	`Alt+Up` or `Alt+Down`	Use these buttons to move the selected configuration or group of configurations (folder) up and down in the list. 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.
📁		Use this button to create a new folder. 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. Move run/debug configurations to a folder using drag-and-drop, or ⬆ ⬇ buttons.

Common options

Item	Description

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.
Defaults	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.
Share	Select this check box to make the run/debug configuration available to other team members. If the directory-based project format is used, the settings for a run/debug configuration are stored in a separate xml file in the `.idea\runConfigurations` folder if the run/debug configuration is shared and in the `.idea\workspace.xml` file otherwise. If the file-based format is used, the settings are stored in the `.ipr` file for shared configurations or in the `.iws` file for the ones that are not shared. This check box is not available when editing the run/debug configuration defaults.
Single instance only	If this check box is selected, this run/debug configuration cannot be launched more than once. Every time a new run/debug configuration is launched, IntelliJ IDEA checks the presence of the other instances of the same run/debug configuration, and displays a confirmation dialog box. If you click **OK** in the confirmation dialog box, the first instance of the runner will be stopped, and the next one will take its place. This make sense, when usage of certain resources can cause conflicts, or when launching two run/debug configurations of the same type consumes too much of the CPU and memory resources. If this check box is not selected, it is possible to launch as many instances of the runner as required. So doing, each runner will start in its own tab of the Run tool window.

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	Click this icon to add a task to the list. Select the task to be added: ■ **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 Another Configuration.** Select this option to have another run/debug configuration executed. In the dialog that opens, select the configuration to be run. ■ **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.

Item	Item	Keyboard shortcut	Description Description
		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.
	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:

- Testing
- Creating and Editing Run/Debug Configurations

Procedures:

- Dart Support

External Links:

- https://www.dartlang.org/articles/dart-unit-tests/

Web Resources:

- Developer Community