

Import from Gradle. Page 2

This page displays the structure of a Gradle project, imported into IntelliJ IDEA. For each of the modules comprising the project, you can see its content root and dependencies (if any). The content roots are not editable.

Depending on the selected node, you can see the following settings:

- [Project](#)
- [Modules](#)
- [Content roots](#)

Project

Item	Description
Name	This field by default shows the name of the imported Gradle project. You can accept default, or change the project name as required.
Language level	Select the Java language level to be supported. The selected level will be used as the project default. The available options correspond to JDK versions: <ul style="list-style-type: none">■ 1.3 - Plain old Java■ 1.4 - 'assert' keyword■ 5.0 - 'enum' keyword, autoboxing, etc.■ 6.0 - @Override in interfaces■ 7.0 - Diamonds, ARM, multi-catch, etc.■ 8.0 - Lambdas, type annotations <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">An individual language level may be set for any of your modules.</div>
SDK	Use this group of controls to define the default SDK for the project. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">You can define an individual SDK for each module.</div>
Project config location	Specify here the path to the directory, where <code>.idea</code> folder should reside. This field by default shows the project root directory, specified in the previous page.

Item	Description
Project compile output location	<p>Specify the path to the directory in which IntelliJ IDEA will store the compilation results. Click ellipsis button to select the directory in the Select Path dialog.</p> <p>Two subdirectories in the specified directory will be created:</p> <ul style="list-style-type: none"> ■ production for production code. ■ test for test sources. <p>In these subdirectories, individual output directories will be created for each of your modules.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>The output paths may be redefined at module level.</p> </div>

Modules

Item	Description
Name	This field by default shows the module name of the imported Gradle project. You can accept default, or change the module name as required.
Module file location	Specify here the directory where *.iml file for the selected module will be stored. By default, this is the Project location directory.
Module config location	Specify here the directory where *.iml file for the selected module will be stored. By default, this is the Project location directory.
Inherit project compile output path	Click this radio button to direct the compiler output to the path, configured on the project level .
Use module compile output path	Click this radio button to direct compiler output to the specified location: <ul style="list-style-type: none"> ■ Output path: specify the directory for the production classes. ■ Test output path: specify the directory for the test classes.
The following fields appear for the dependencies only.	
Export	<p>If this check box is selected, a dependency is included in the compilation classpath of the dependent modules, as well as the classpath of the module which it is attached to.</p> <p>If this check box is not selected, a dependency is only included in the compilation classpath of a module which it is attached to. Dependencies are always included in the runtime classpath regardless of the state of the Export check box.</p>

Item	Description
Scope	<p>Use this drop-down to affect the classpath for the various build phases.</p> <ul style="list-style-type: none">■ <i>Compile</i>: This is the default option. If it is selected, the dependency is resolved and is available in classpath during the compilation and run phases.■ <i>Test</i>: Select this option, if this dependency is only required for tests, and should not be available in normal application use. If this scope is selected, the dependency is resolved and is available in classpath during the test compilation and run phases.■ <i>Runtime</i>: This scope indicates that the dependency is only required when running the application, and should not be available in classpath during the compilation.■ <i>Provided</i>: If this option is selected, the dependency is resolved and is available in classpath during the compilation, but is not included in classpath at runtime. This dependency scope is useful, when you have some container that provides the dependency at runtime.

Content roots

The fields for the content roots are not editable. Use this pane to review the [source, test and excluded roots](#).

See Also

Language and Framework-Specific Guidelines:

- [Importing Project from Gradle Model](#)

Reference:

- [Gradle](#)

Web Resources:

- [Developer Community](#) 