Module Category and Options

In the left-hand pane, select the module category. This may be the technology that you are going to use, the platform or runtime that your development is going to target, etc.

In the right-hand part of the page, select additional options and specify associated settings.

Note that the set of options you can select from depends on which plugins are currently enabled in IntelliJ IDEA.

- Java
- Java Enterprise
- Spring
- J2ME
- IntelliJ Platform Plugin
- Android
- Maven
- Gradle
- Groovy
- Static Web
- Flash

Java

Select this option if you are going to use your new module to develop a Java 🗗 application.

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and select the SDK type. (It's likely that it's the **JDK** option that you want at the moment.) Then, in the dialog that opens, select the installation folder of the desired SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)

If necessary, select additional options and specify associated settings. For more information, see Additional Technologies.

Java Enterprise

Select this option if you are going to use your new module to develop a Java EE 🗗 application.

Item	Description

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and select the SDK type. (It's likely that it's the **JDK** option that you want at the moment.) Then, in the dialog that opens, select the installation folder of the desired SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)
Java EE version	Select the Java EE version to be supported.
Application Server	Specify the application server that you are going to use to deploy and run your application. As a result, IntelliJ IDEA will create a run/debug configuration for the specified server. (You can specify the server later.) You can select a server which IntelliJ IDEA is already aware of, or specify another "new" server. To specify a new server, click **New** and select the server of interest. Then, specify the server settings: ■ For a server installed locally, specify the path to the server installation directory. (Click ⚏ to select the directory in the corresponding dialog.) ■ For a hosted server (Cloud Foundry or CloudBees), specify your user account details.

Select additional options and specify associated settings. For more information, see Additional Technologies.

Spring

Select this option if you are going to use your new module to develop a Spring ⬈ application.

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and select the SDK type. (It's likely that it's the **JDK** option that you want at the moment.) Then, in the dialog that opens, select the installation folder of the desired SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)

For information on other options and settings, see:

- Spring
- Spring MVC, Spring Batch, or other Spring framework
- Additional Technologies

See also, Enabling Spring Support.

J2ME

Select this option if you are going to use your new module to develop for Java ME.

Item	Description
SQL Support	Select the check box to enable SQL support. Select the SQL dialect to be used by default from the list.

IntelliJ Platform Plugin

Select this option if you are going to use your new module to develop a plugin for IntelliJ IDEA or other IntelliJ Platform-based IDE.

Item	Description
Module SDK	Specify an SDK to be used. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and, in the dialog that opens, select the installation folder of the desired IntelliJ IDEA version. (An IntelliJ IDEA installation acts as an IntelliJ IDEA Plugin SDK.) (By this time, the corresponding IntelliJ IDEA version must already be installed on your computer. If it isn't, download and install it first.)
Groovy	Select the check box to be able to use Groovy. Specify the Groovy installation to be used. **Use library.** If the desired version of Groovy is already defined in IntelliJ IDEA, select it from the list. (Groovy in IntelliJ IDEA is represented by a library.) **Create.** Click this button to create a library for Groovy. In the dialog box that opens, select the Groovy installation directory.
SQL Support	Select the check box to enable SQL support. Select the SQL dialect to be used by default from the list.

Android

Select this option if you are going to use your new module to develop for the Android OS. Select:

- **Application Module** to develop an Android application.
- **Empty Module** to develop an Android application or library.
- **Library Module** to develop a shared Android library.
- **Test Module** to develop tests for your Android application (represented by an Android application module). See the documentation for the Android testing framework used.

Maven

Select this option if you are going to use your new module to develop a Java ⧉ application with dependencies managed by Maven ⧉.

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and, in the dialog that opens, select the installation folder of the desired Java SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)
Create from archetype	If this check box is not selected, the new `pom.xml` file will contain the basic information. If this check box is selected, the new module will be created on the base of a Maven archetype chosen from the list that includes both the standard archetypes, and the ones found in Maven indices. You can modify Maven properties on Maven Settings Page. If you want to populate the list with some archetype from a remote Maven repository, click the **Add Archetype** button, and find the desired archetype by Maven coordinates specified in Add Archetype Dialog.

Gradle

Select this option if you want a module with a Gradle build script (`build.gradle`) to be created.

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and select the SDK type. (It's likely that it's the **JDK** option that you want at the moment.) Then, in the dialog that opens, select the installation folder of the desired SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)

Groovy

Select this option if you are going to develop a Groovy ⧉, Grails ⧉ or Griffon ⧉ application.

Item	Description

Item	Description
Module SDK	If IntelliJ IDEA can guess which SDK you are likely to use in your module, the controls for specifying the module SDK are not provided. This may be the case when an SDK of the appropriate type is defined as the project SDK, or there is already a module in your project where an appropriate SDK is used, etc. Otherwise, you can specify the SDK to be used in your module. If the necessary SDK is already defined in IntelliJ IDEA, select it from the list. Otherwise, click **New** and select the SDK type. (It's likely that it's the **JDK** option that you want at the moment.) Then, in the dialog that opens, select the installation folder of the desired SDK. (By this time, the corresponding SDK must already be installed on your computer. If it isn't, download and install it first.)
Groovy library	If the desired version of Groovy is already defined in IntelliJ IDEA, select it from the list. (Groovy in IntelliJ IDEA is represented by a library.) **Create**. Click this button to create a library for Groovy. In the dialog box that opens, select the Groovy installation directory.

Static Web

Select **Static Web** if you are going to develop a Web application using HTML/CSS, JavaScript, PHP and related frameworks.

The other options are for developing a Web site or a Web front end using an HTML5 Boilerplate template, or the Foundation or the Bootstrap framework.

Flash

Select this option if you are going to use your new module to develop for the Adobe Flash runtimes using Flex or ActionScript . Specify the associated settings.

Item	Description
Target platform	Select the target environment for the content that you are going to develop: - **Web** for Flash player / Web browser-targeted content. - **Desktop** for Adobe AIR-targeted content. - **Mobile** for the content intended for mobile devices (Android, iOS, etc.).
Pure ActionScript	Select this check box if you are not going to use MXML (i.e. all your source code will be written in ActionScript).
Output type	Select the intended output type, that is, what your resulting content is going to be: - **Application.** A runnable application, an SWF file. - **Runtime-loaded module.** A dynamically-loadable module, an SWF file. - **Library.** An SWC file.

Item	Description
Target devices	For a Mobile Application: use the **Android** and **iOS** check boxes to specify the intended target devices for your application. As a result, IntelliJ IDEA enables or disables creating an application descriptor and packaging your application for the corresponding devices. (The **Android** and **iOS** check boxes on this page correspond to the **Enabled** check boxes on the Android and iOS tabs in the build configuration that will be created.)
Flex/AIR SDK	Select the Flex or AIR SDK to be used. If the list is empty or does not contain the required SDK, click ⊡ (Shift+Enter) and add the required SDK in the **Configure SDK** dialog.
Target player	For the Web target platform: the target Flash Player version (readonly). (This setting is defined by the selected Flex SDK version.)
Create sample app	For the Application output type: select this check box if you want a sample application to be created. You can use this sample application for learning and also as a basis for your own application development. If necessary, change the source file name suggested by IntelliJ IDEA.
Create HTML wrapper template	For a Web Application: select this check box if you want an HTML wrapper ⧉ template for your application to be created. Select or deselect the associated options as needed: ■ **Enable integration with browser navigation .** Select this option to enable deep linking. Deep linking lets users navigate their interactions with the application by using the **Back** and **Forward** buttons in their browser. ■ **Check Flash player version .** If you select this option, the compiled application will check for the correct version of Flash Player. ■ **Express install .** If you select this option, the application will run an SWF file in the existing Flash Player to upgrade users to the latest version of the player.