Play Framework

IntelliJ IDEA supports the version 1.x.x of the Play framework. The necessary integration with the framework is provided by the Playframework Support plugin bundled with with the IDE.

A Play application in IntelliJ IDEA corresponds to a Java module.

> If you want to use IntelliJ IDEA just for running the `play` command-line utility (also known as the Play console), you can install and use the version 2.x.x of the Play framework.

On this page:

Play framework support overview

The Play framework support in IntelliJ IDEA includes:

- An ability to use the `play` command-line utility (the Play console) right in the IDE.
- Coding assistance:
  - Navigation between the methods of `Application.java` and the corresponding templates.
  - Updating templates through the variables of the corresponding `Application.java` methods.
  - Code completion for keywords, labels, variables, parameters and functions.
  - Error and syntax highlighting.
  - Code formatting and folding.
- Advanced source code search and replace capabilities.
- Structure view.

Preparing for Play application development

To prepare for Play application development:

1. Download and install the desired Play framework version supported by IntelliJ IDEA. See the installation instructions.

2. Create a new Play application (`play new`) or get an existing one (the one that you are going to work with in IntelliJ IDEA).
3. Prepare your application for opening it in IntelliJ IDEA (`play idea` or `play idealize`).
4. Make sure that the Play framework settings are properly specified in IntelliJ IDEA.
5. Open the application in IntelliJ IDEA.
6. Optionally, convert the project into the directory-based format. See Converting Projects Into Directory-Based Format.

Creating a Play application

To create a new Play application, run the `play new` command in a command-line shell, outside IntelliJ IDEA:

1. Open your command-line shell.
2. Switch to the directory in which you want to create your Play application.
3. Run the following command:

   ```
   play new <dir>
   ```

   where `<dir>` is the name of your new Play application directory (e.g. `helloworld`) which will be created in the current directory.
4. When asked for the application name, type the name and press **Enter**.

As a result, the Play application with the specified name is created in the specified directory.

Preparing a Play application for opening it in IntelliJ IDEA

To prepare a Play application for opening it in IntelliJ IDEA, run the `play idea` or the `play idealize` command from the root directory of your Play application:

1. Open your command-line shell.
2. Switch to the root directory of your Play application.
3. Run the following command:

   ```
   play idea
   ```

   (Alternatively, you can use the `play idealize` command.)

As a result, all the necessary IntelliJ IDEA configuration files are created in the current directory. These include the `<app_name>.ipr` project file which you can now open in IntelliJ IDEA.

Specifying Play framework settings

In IntelliJ IDEA, the Play framework settings are specified in the **Settings** dialog, on the **Play Configuration** page.

1. Open the **Settings** dialog (e.g. **Ctrl+Alt+S**).
2. In the left-hand pane of the dialog, select **Play Configuration**.
3. On the Play Configuration page that opens in the right-hand part of the dialog:
   - In the **Home** field, specify the Play framework installation directory.
   - In the **Working directory** field, specify the Play framework working directory. This is the directory from which the commands of the `play` command-line utility are run.
   - Depending on your preferences, turn the Show on console run option on or off.
4. Click **OK** in the **Settings** dialog.

Opening a Play application in IntelliJ IDEA

To open a Play application in IntelliJ IDEA, you should open the `<app_name>.ipr` project file you have previously generated:

1. Select **File | Open**.
2. In the **Open File or Project** dialog, go to your Play application root folder, select the `<app_name>.ipr` file, and click **OK**.

In the project that opens your Play application is represented by an IntelliJ IDEA module.

If necessary, convert your project into the directory-based format. See Converting Projects Into Directory-Based Format.

An alternative way to create an IntelliJ IDEA project for a Play application

The quickest and the most convenient way to start working with a Play application in IntelliJ IDEA is to run the play idea command and then open the generated .ipr file in IntelliJ IDEA.

As an alternative, you can create a new project for your Play application sources using **File | Import Project | Create project from existing sources** and then add the necessary Play framework assets (`<play_dir>\framework\lib` and `<play_dir>\framework\play-<version>.jar`) to dependencies of the resulting IntelliJ IDEA module:

1. Select **File | Import Project**.
2. In the dialog that opens, select your Play application root directory.
3. On the first page of the **Import Project** wizard, select **Create project from existing sources** and click **Next**.
4. On the next page of the wizard, in the **Project location** field, specify the path to your Play application root directory. The rest of the settings are not that important. Click **Next**.
5. Follow the instructions of the wizard. (Normally, all you have to do is to click **Next** on each of the pages accepting the default settings.)

   When the project has been created, add the necessary module dependencies:

6. Open the Project Structure dialog (e.g. **Ctrl+Shift+Alt+S**).
7. In the left-hand pane of the dialog, select **Modules**.
8. On the Module page shown in the right-hand part of the dialog, select the Dependencies tab, click + and select **Jars or directories**.
9. In the **Attach Files or Directories** dialog, go to the Play framework installation directory, select the directory `framework\lib` and the file `framework\play-<version>.jar`, and click **OK**.
10. Click **OK** in the **Project Structure** dialog.

Accessing the play command-line utility in IntelliJ IDEA

If a Play application is currently open in IntelliJ IDEA, you can access the `play` command-line utility (the Play console) and run it right from the IDE:

1. Select **Tools | Play with Playframework**.
2. In the Play Configuration dialog specify the Play framework settings and click **OK**. (This dialog is not shown if the Show on console run option is off in the Play framework settings.)

As a result, the `play` command-line utility is started in the Run tool window.

Adding Play modules to an IntelliJ IDEA module

The Play modules used by your Play application should be added to the corresponding IntelliJ IDEA module as the module content roots. For example, if your application uses (or is about to be using) the Play module `secure`:

1. Open the Project Structure dialog (e.g. **Ctrl+Shift+Alt+S**).
2. In the left-hand pane of the dialog, select **Modules**.
3. On the Module page shown in the right-hand part of the dialog, select the Sources tab and click **Add Content Root**.
4. In the **Select content root directory** dialog, go to the Play framework installation folder, select the directory `modules\secure`, and click **OK**. (If the Play module you want isn't included in the Play framework distribution, you should download that module first to make it available locally.)
5. Click **OK** in the **Project Structure** dialog.

Running a Play application

To run your play application, you can use the `play run` command:

1. Start the play command-line utility (**Tools | Play with Playframework**).
2. In the **Run** tool window, after `play`, type `run` and press **Enter**.
3. Open a Web browser and go to `http://localhost:9000` to see the application home page.

As an alternative, you can create an IntelliJ IDEA run/debug configuration, and use that configuration for running your application.

To create a run configuration for running your Play application:

1. Open the **Run/Debug Configurations** dialog (e.g. **Run | Edit Configurations**).
2. Click + (**Alt+Insert**) and select **Application**.
3. In the following fields, specify:
   - **Main class.** Type `play.server.Server`
   - **VM options.** Type `-Dapplication.path="."`
   - **Working directory.** Specify your Play application root directory.
4. If necessary, change the run configuration name and click **OK**.

Now, to execute this run configuration, you can, for example, use **Run | Run**. (For more information, see Running Applications.)

Running Play application tests

To run tests for your Play application, you can use the `play test` command:

1. Start the play command-line utility (**Tools | Play with Playframework**).
2. In the **Run** tool window, after `play`, type `test` and press **Enter**.
3. Open a Web browser and go to `http://localhost:9000/@tests` to run the tests.

To be able to use an IntelliJ IDEA run/debug configuration for running your tests:

- Add `<play_dir>\modules\testrunner\lib\play-testrunner.jar` to the dependencies of your module. (See an example of the procedure to be used.)
- Modify the run configuration for running the application: in the **VM options** field, after `-Dapplication.path="."`, type space and then type `-Dplay.id=test`

  So, finally, the **VM options** field will contain:

  ```
  -Dapplication.path="." -Dplay.id=test
  ```

  (If you want a separate run configuration for the tests, you can create a copy of the existing run configuration ( ) and then modify that copy accordingly.)

Debugging a Play application: process overview

1. Create a run/debug configuration for debugging. This should be a **Remote** type of configuration; the port specified in this configuration should correspond to a Java debugger port (`8000` by default).
2. Set one or more breakpoints in your code. See Using Breakpoints.
3. If you are going to use an IntelliJ IDEA run configuration for starting the application, modify the corresponding run configuration, or create a new one.
4. Run the application by using the `play run` command (without any options and arguments) or by executing the corresponding run configuration.
5. Start the run/debug configuration intended for debugging.
6. In a Web browser, go to `http://localhost:9000` and, by using the application, try to reach a breakpoint.
7. When a breakpoint is reached, switch to IntelliJ IDEA and scrutinize the suspicious code fragments.
8. Continue debugging unless you localize the problem in your code.

See also, Debugging.

Creating a run/debug configuration for debugging

1. Open the **Run/Debug Configurations** dialog (e.g. **Run | Edit Configurations**).
2. Click + (**Alt+Insert**) and select **Remote**.
3. In the **Port** field, specify `8000`. (This is the port to which a Java debugger will connect. By default, the Play framework uses the port `8000`.)
4. If necessary, change the run/debug configuration name and click **OK**.

Modifying the run configuration for running the application

If you are using an IntelliJ IDEA run configuration for running your Play application, this configuration has to be modified so that a Java debugger could connect to the running application. This is done by adding the corresponding JVM options. The necessary options can be copied from the run/debug configuration intended for debugging.

1. Open the **Run/Debug Configurations** dialog (e.g. **Run | Edit Configurations**).
2. In the left-hand pane of the dialog, select the run/debug configuration intended for debugging.
3. Click ⎘ to the right of the upper field containing JVM command-line arguments. As a result, the field contents are copied to the clipboard.
4. Select the run configuration for running the application, and paste the options into the **VM options** field. This field should finally contain `-Dapplication.path="."`, then a space and then the options you've just pasted. Here is an example:

   ```
   -Dapplication.path="." -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000
   ```

   (If you are using the same run configuration for running the application and the tests, you may need to delete `-Dplay.id=test`.)
5. Click **OK**.

See Also

Reference:

- Play Configuration
- Play Configuration Dialog

- Play Framework (Play Console)

External Links:

- Play framework

Web Resources:

- Developer Community