IDE

Table of contents

Creating a new IsFusion project

Launch **IDEA** and select *Create New Project*, or when **IDEA** is already opened, select *File* -> *New* -> *Project* from the menu.
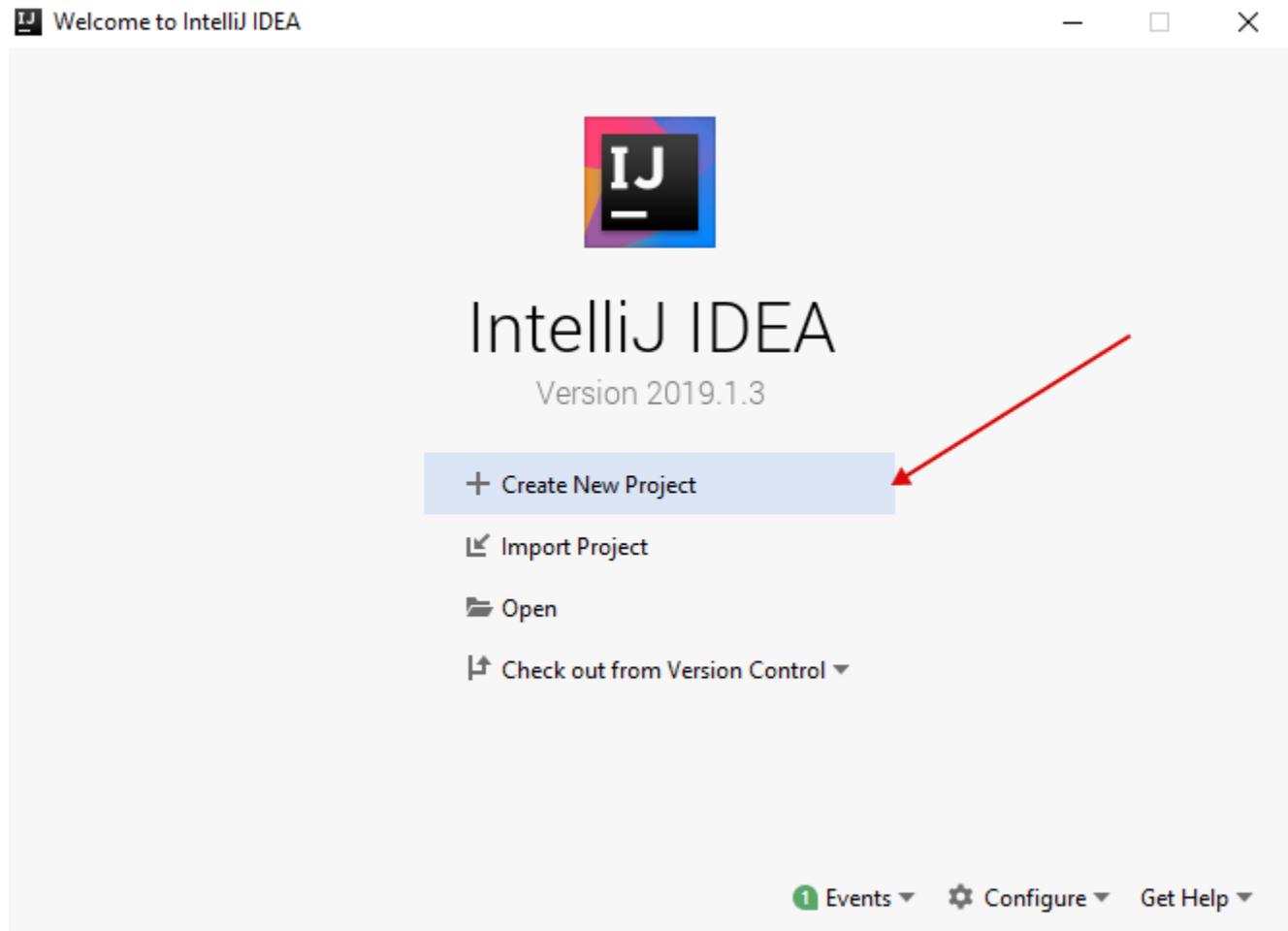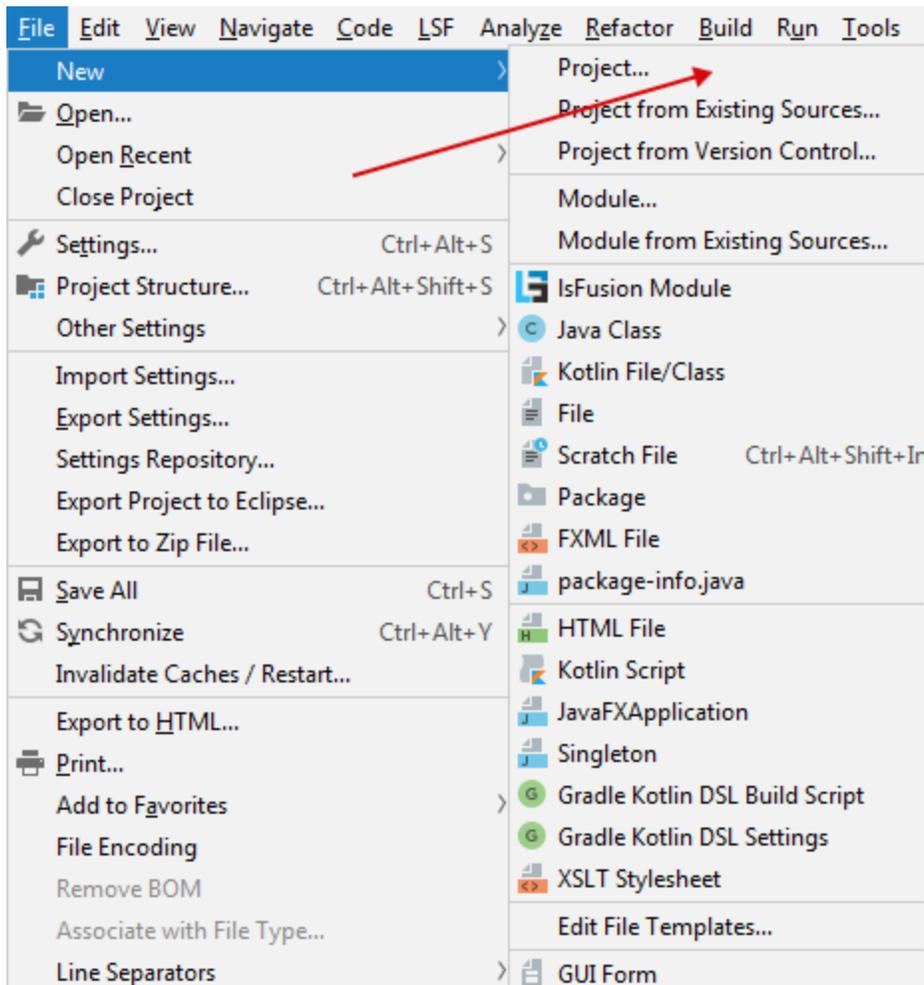
Select project type **IsFusion**. Make sure that the JDK is set and the link to the library with the **IsFusion** is selected (when automatic installation is used, this link is set automatically; otherwise, click *Download* or select the previously downloaded library *Create*) and then click *Next*.
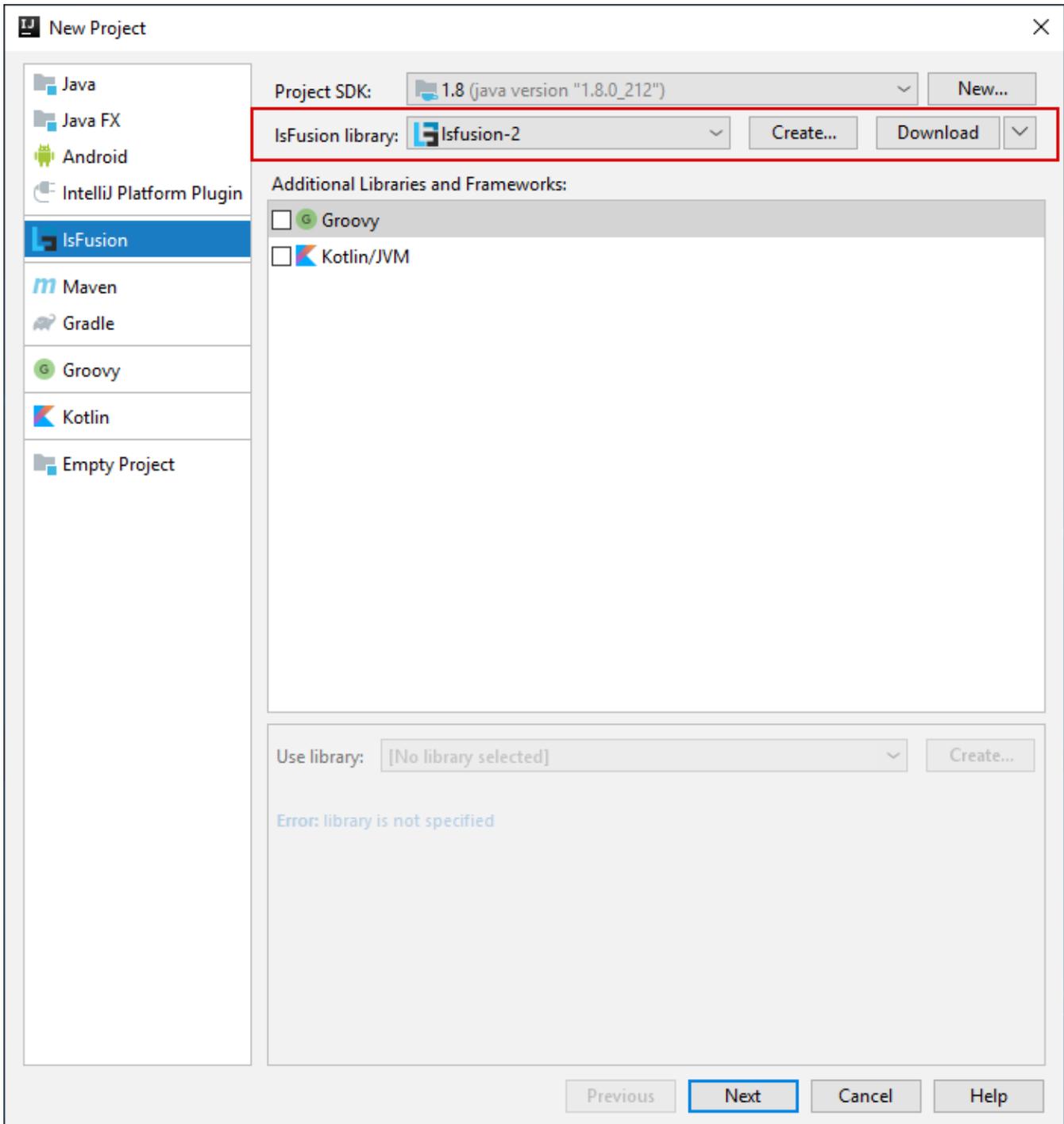
Fill in the name of the project and the directory where the source files will be stored. Adjust the connection parameters for the database and application server when necessary. Click *Finish*.

Developing an application

All the source code written in **lsFusion** is stored by default in src/main/lsfusion.

To add a new module, right-click the corresponding folder and select *New* / *lsFusion* from the menu:
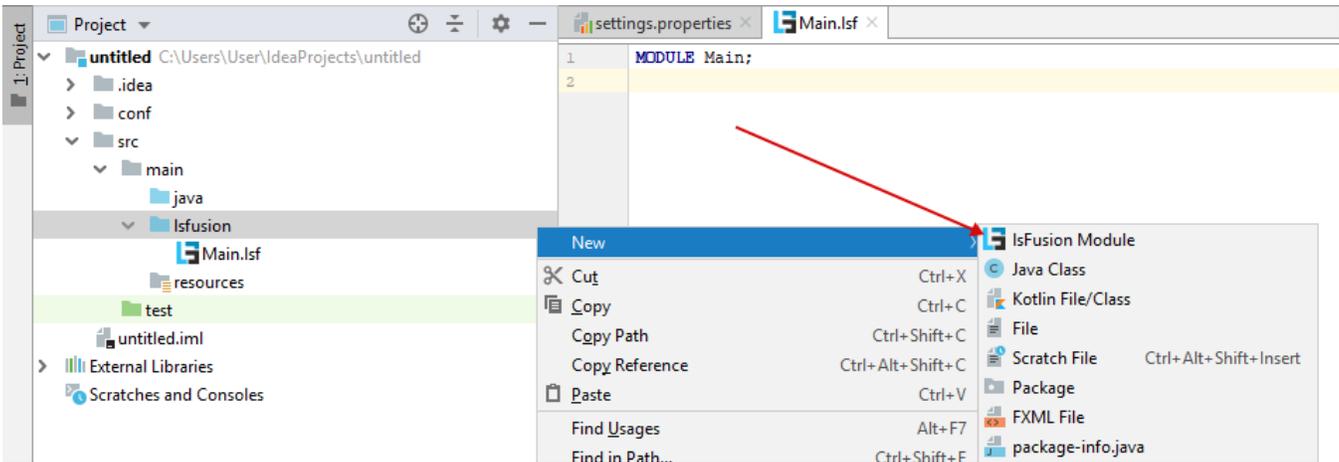
You can also create subfolders (by choosing *Package*) to group similar modules into a directory.

Starting up a server

When you create a new project, a server startup configuration is also created by default. You can run it by selecting "Run -> Run 'Run lsFusion server,'" or by clicking on the symbol ▶ next to the caption *Run lsFusion server* in the upper right corner. If startup is successful, the last line in the log should be *Server has successfully started*.
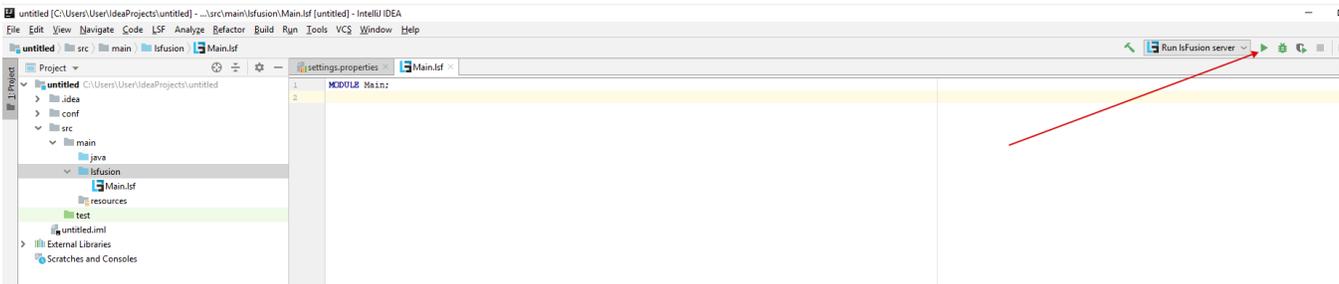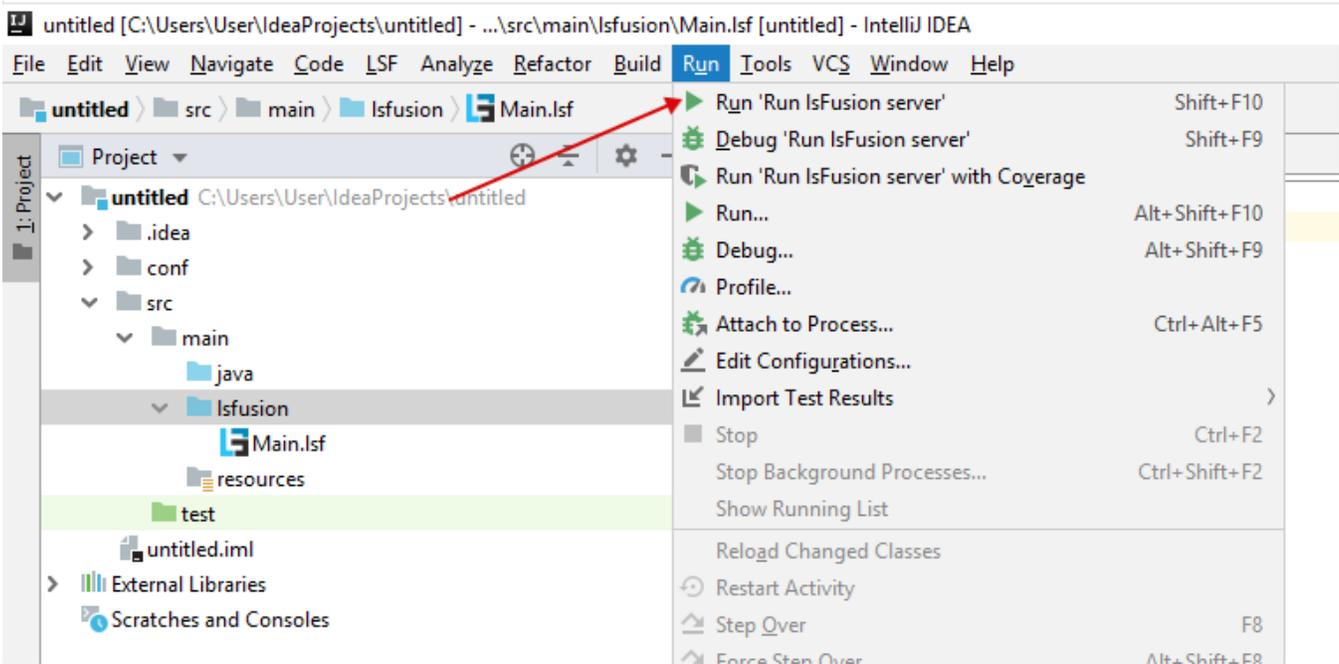
After the server has been successfully started, you can connect to it over a web-based or desktop client via a shortcut on your desktop.

Creating a server startup configuration

If the platform is embedded to an existing project (for example, via Maven for a Maven project), you may need to manually create a server startup configuration for the applications. This is done as follows:

In the menu, select *Edit configurations*:

In the window that opens, click + in the upper left corner and select *IsFusion Server* from the drop-down list
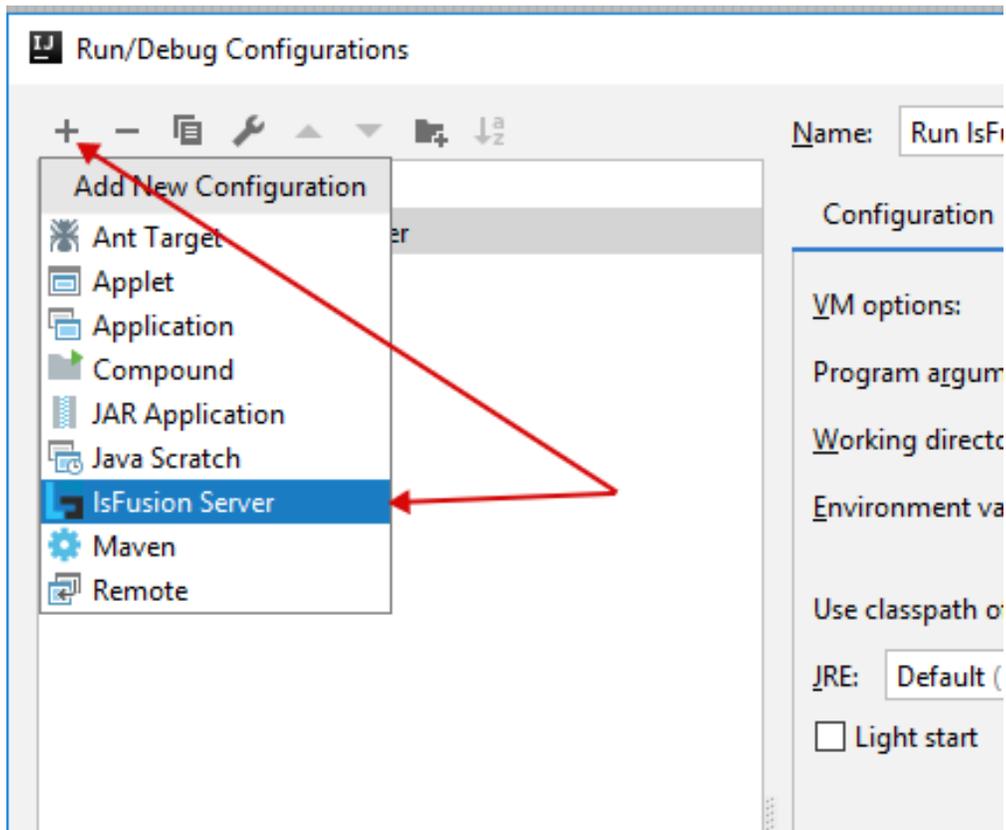
Set the configuration name and the module (if there are several) for which you need to start the IsFusion application server. In the same window, if necessary, you can specify the build process to be followed at configuration startup, additional parameters for the Java virtual machine, etc.
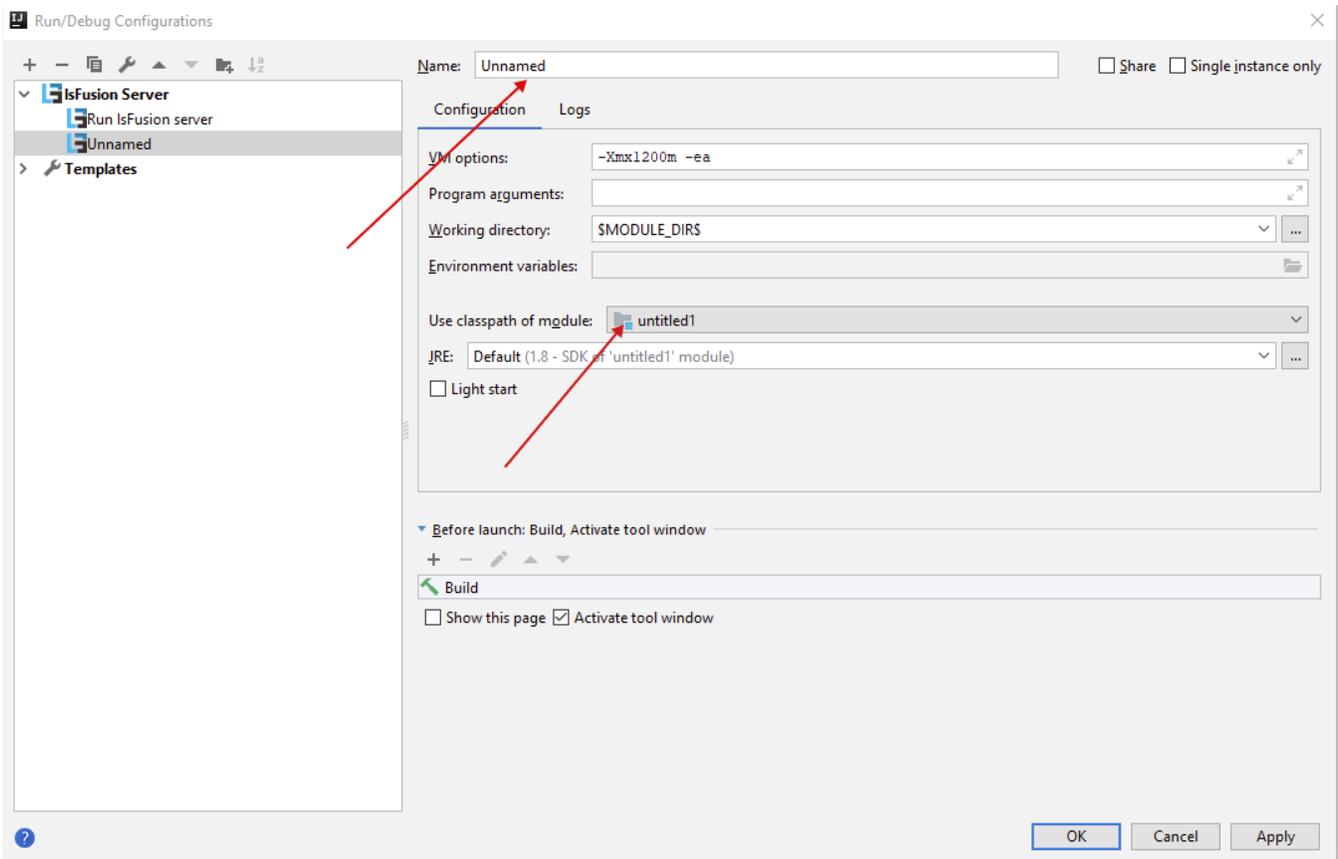
Building an application (with embedded server)

To compile a single JAR file that contains both the developed code and the application server itself, you can use the following guide. The result JAR file (*art ifact*) can be used to install applications on a production server as described in the section installing an applications server as a service. Note that since all modules and the server itself will be inside the same JAR file, the installation process is slightly different:

- in the first section instead of the server JAR file this file must be downloaded
- the third section (copying application files to the server) can be skipped