

Context and Dependency Injection (CDI)

The [Context and Dependency Injection](#) (CDI) support in IntelliJ IDEA is based on the CDI Support [plugin](#). This plugin is bundled with the IDE and enabled by default.

You can enable CDI support when creating a [project](#) or [module](#). You can as well add CDI support for an existing Java module.

CDI in a module is represented by the corresponding CDI [library](#) in the [module dependencies](#). You can download all the necessary library files right from within the IDE. You can also change the CDI version being used if and when needed.

- [Making sure that the CDI Support plugin is enabled](#)
- [Enabling CDI support when creating a project or module](#)
- [Adding CDI support for an existing module](#)
- [Changing the CDI version](#)

Making sure that the CDI Support plugin is enabled

Even though the CDI Support plugin is enabled by default, it's always worth making sure that this plugin is still enabled before you start using CDI.

1. [Open the Settings dialog](#) (e.g. `Ctrl+Alt+S`).
2. In the left-hand part of the dialog, under **IDE Settings**, select **Plugins**.
3. In the right-hand part of the dialog, on the [Plugins page](#), type `cd` in the search box. As a result, only the plugins whose names and descriptions contain `cd` are shown in the list of plugins.
4. If the check box to the left of **CDI Support** is not selected, select it.
5. Click **OK** in the **Settings** dialog.
6. If suggested, restart IntelliJ IDEA.

Enabling CDI support when creating a project or module

Depending on the situation, you can choose to create a new project with a Java module or to add a Java module to an existing project.

1. Do one of the following:

- If you are going to create a new project: click **Create New Project** on the [Welcome screen](#) or select **File | New Project**.

As a result, the [New Project wizard](#) opens.

- If you are going to add a module to an existing project: [open the project](#) you want to add a module to, and select **File | New Module**.

As a result, the [New Module wizard](#) opens.

2. On the first page of the wizard, in the left-hand pane, select **Java**.

3. In the right-hand part of the page, specify the associated settings and click **Next**. (See the descriptions of the settings [for a new project](#) or [for an existing project](#).)

4. On the next page of the wizard (the [Technologies page](#)), in the left-hand pane, select the **CDI: Context and Dependency Injection** check box.

5. In the right-hand part of the page, select the required library option and, if necessary, specify the associated settings. You can choose to:

- Download and use a CDI implementation library.

To do that, select **Download**.

Now, to view or modify the associated options, click **Configure**, and in the **Downloading Options** dialog that opens:

- Select the CDI version that the library should implement.
- Specify the library name.
- Select the [library level](#) (global, project, or module).
- Under **Files to download**, select which of the files you want to download.
- Under **Copy downloaded files to**, specify the path to the destination folder. If you want to change the default path, click  and specify the folder location in the [dialog that opens](#).
- Use a CDI library IntelliJ IDEA is already aware of.

To do that, click **Use library** and select the required library from the list.

If necessary, configure the library settings (for example, change its name). This is done in the **Edit Library** dialog which you can open by clicking **Configure**.

- Create a new library using the appropriate JAR files available on your computer.

To do that, click **Use library** and then click **Create**. Select the required JAR files in the [dialog that opens](#). (Keep the Ctrl key pressed to perform multiple selection.)

If necessary, configure the new library (for example, change its name or [level](#)). To do that, click **Configure** and specify the required settings in the **Create Library** dialog.

6. Click **Finish**.

Adding CDI support for an existing module

1. [Open the module settings](#).
2. With the module of interest selected, click **+** on the toolbar, and, under **Framework**, select **CDI: Context and Dependency Injection**.
3. In the dialog that opens, select the required library option. If necessary, specify the associated settings, and click **OK**. You can choose to:

- Download and use a CDI implementation library.

To do that, select **Download**.

Now, to view or modify the associated options, click **Configure**, and in the **Downloading Options** dialog that opens:

- Select the CDI version that the library should implement.
- Specify the library name.
- Select the [library level](#) (global, project, or module).
- Under **Files to download**, select which of the files you want to download.
- Under **Copy downloaded files to**, specify the path to the destination folder. If you want to change the default path, click  and specify the folder location in the [dialog that opens](#).
- Use a CDI library IntelliJ IDEA is already aware of.

To do that, click **Use library** and select the required library from the list.

If necessary, configure the library settings (for example, change its name). This is done in the **Edit Library** dialog which you can open by clicking **Configure**.

- Create a new library using the appropriate JAR files available on your computer.

To do that, click **Use library** and then click **Create**. Select the required JAR files in the [dialog that opens](#). (Keep the **Ctrl** key pressed to perform multiple selection.)

If necessary, configure the new library (for example, change its name or [level](#)). To do that, click **Configure** and specify the required settings in the **Create Library** dialog.

As a result, the CDI library you have specified is added to the list of module dependencies.

Changing the CDI version

1. [Open the Project Structure dialog](#) (e.g. **Ctrl+Shift+Alt+S**).
2. Select the CDI library whose version you want to change.
3. Click **Change Version**. (If you are on the **Dependencies** tab, select to edit the library first.)
4. In the **Downloading Options** dialog that opens, select the necessary CDI version, specify other settings as needed and click **OK**.

See Also

Reference:

- [CDI Tool Window](#)

Web Resources:

- [Developer Community](#) 