Creating Template-based Files

IntelliJ IDEA provides file templates for most of the languages that it supports. This lets you create the files with the initial content appropriate for the file purpose. For example, there are individual file templates for creating Java classes, interfaces, enumerations, annotations, scripts, GUI forms, etc.

Generally, the file name extension for a template-based file is set automatically so you don't need to specify it. This, for example, is the case when creating Java types (classes, interfaces, etc.): the corresponding files always get the .java extension. Similarly, if you create a Groovy class, a new file will get the .groovy extension.

Sometimes, you are given an opportunity to select the desired extension from the list. This, for example, is the case when creating PHP classes.

Note that the set of file templates you can choose from, generally, depends on which facets your module has and, to a lesser extent, on your current location in the Project tool window.

To create a new template-based file

1. Do one of the following:

   - In the Project tool window, select the directory or package in which you want to create a new file, and then choose **File | New** on the main menu.

   - Right-click the corresponding directory or package and select **New** from the context menu.

   - Press Alt+Insert

2. Select the desired file type. Generally, all the options except **File** , **Package** and **Directory** correspond to using a file template.
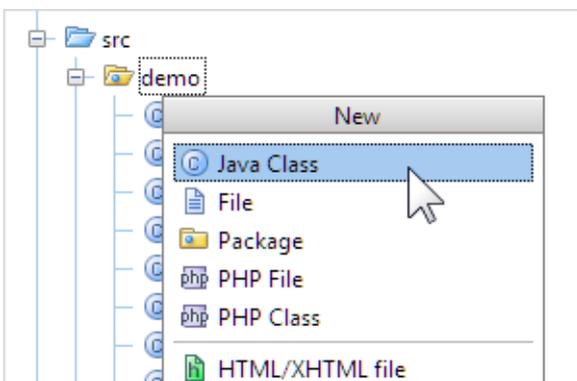

An existing file template may be missing from the list for one of the following reasons:

- The template is associated with a facet that your module doesn't have. In such a case, you may want to add the corresponding facet to your module.

- The corresponding file type is inappropriate for the current location. For example, you may be trying to create a Java class outside of Java source or test directory which is not possible.

- The template you are looking for is a custom template whose file name extension (template extension) does not match registered patterns of any of the recognized file types. In such a case, you may want to register the corresponding pattern for an existing recognized file type or add a new file type and register the corresponding pattern for this new type. For more information, see Creating and Registering File Types.

3. In the dialog that opens, type the name of the file in the corresponding field. Note that you should not type the file name extension.

   You can specify the whole directory structure prepending the new file name. If the nested directories do not yet exist, they will be created:
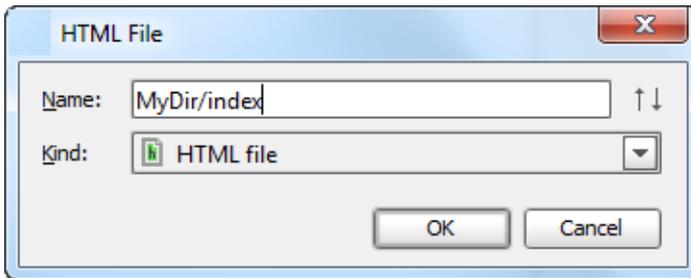

   If required, specify the kind of the new template-based file. For example, if you select *Java class*, you'll be able to create a Java class, interface, enumeration, annotation, etc.; if you select to create an *HTML file*, you'll be able to create HTML, HTML4, or XHTML file. In other words, use one of the corresponding related file templates.

   Specify other information as required. For example, you may be asked to define the values of custom variables if the corresponding file template contains such variables and their values are not currently set.

4. Click **OK**. The new file that corresponds to the selected file template will be created under the target location.

   If the names of non-existent sub-directories were specified before the new file name, the whole structure will be created under the target directory:
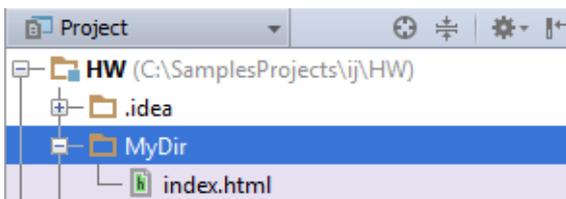
> Sometimes, you may want to change the auto-generated file name extension. To do that, use the Rename refactoring (**Refactor | Rename**).

See Also

Concepts:

- File and Code Templates
- Supported Languages

Procedures:

- Creating and Editing File Templates
- Creating Java Classes, Interfaces, Enumerations and Annotations
- Creating Empty Files
- Creating Node Elements and Members

Reference:

- File and Code Templates

Web Resources:

- Developer Community ⮞