Command Line Tools

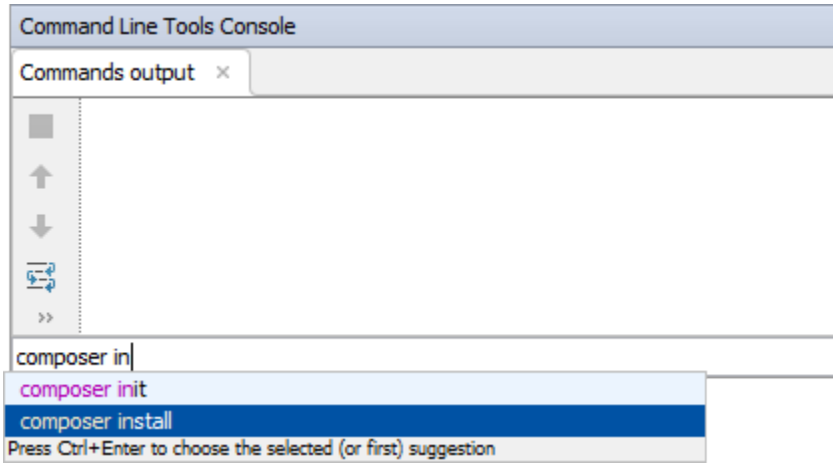
In this tutorial we'll see how to work with command line tools in PhpStorm.

PhpStorm has had support for command line tools for a while. Using the command line tools, we can invoke commands right from our IDE! Before we can use any command line tool, we have to enable it in the settings. Let's go over all the steps involved!

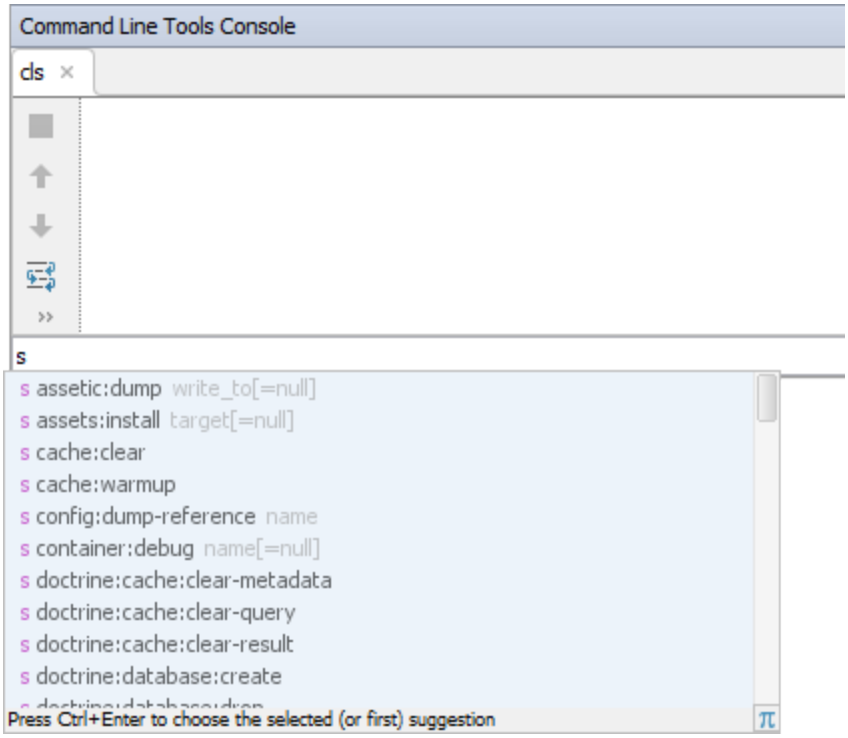
Working with the command line tools console

We can bring up the command line tool using the Tools | Run Command... menu or with Ctrl+Shift+X (Cmd+Shift+X on Mac).

The command line tools come with support for autocompletion of several known tools. Nothing comes for free: we have to add and enable the tools we want to use through the settings. After adding Composer, for example, we get autocompletion on all c omposer commands (note the default alias of "c" has been changed to "composer" in the screenshot below):



We can add other tools as well. The following is an example of working with the Symfony tool, aliased as s.



Enabling command line tools

Enabling a well-known command line tool

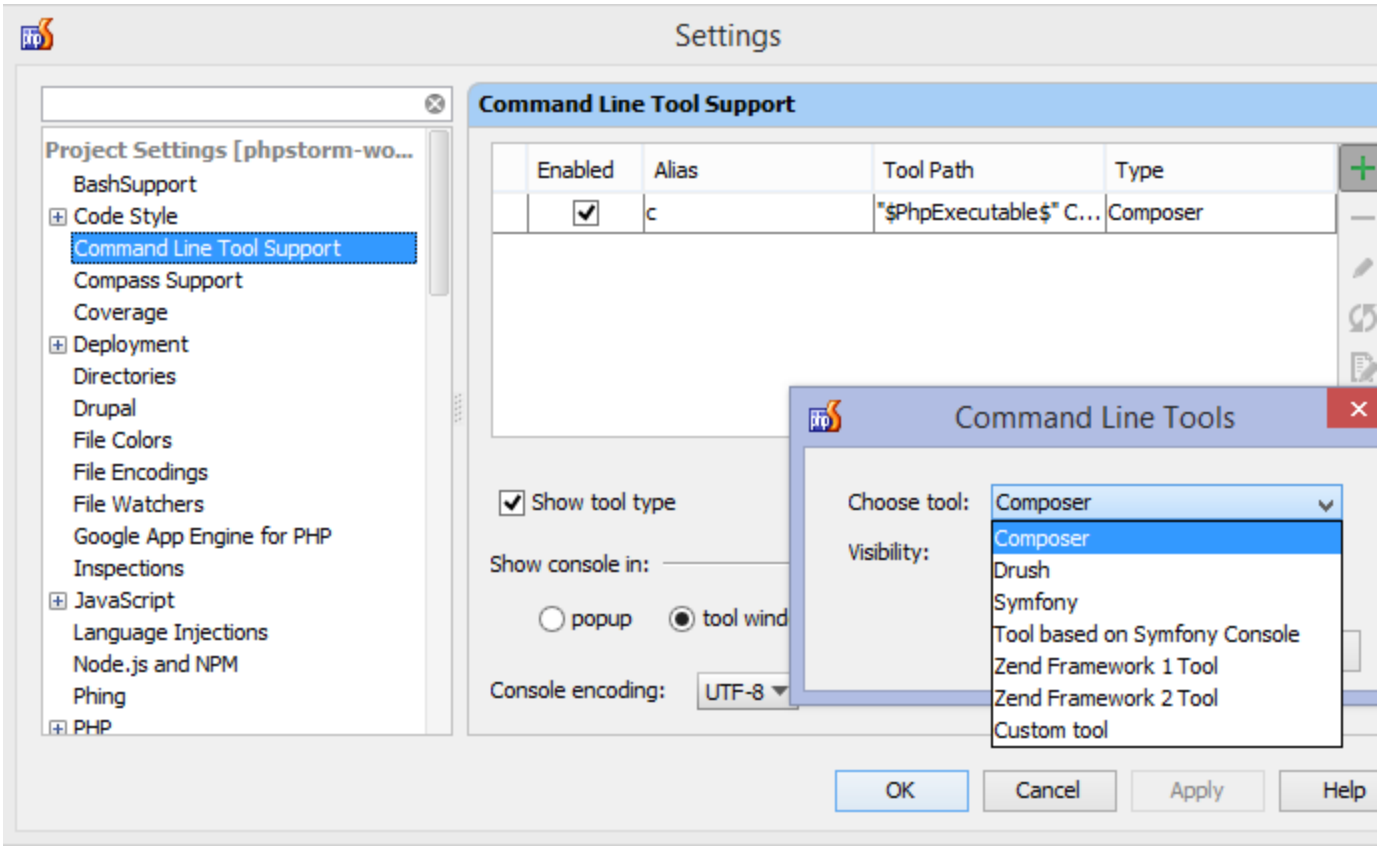
PhpStorm comes with support for various well-known tools:

- Composer
- Zend Framework (version 1 and 2)
- Symfony
- Tools based on Symfony Console (Doctrine, Laravel)
- Drush (for Drupal)

Next to these, custom command line tools are also supported.

Command line tools have to be enabled in the settings. We can do this globally (for all projects) or on a per-project basis.

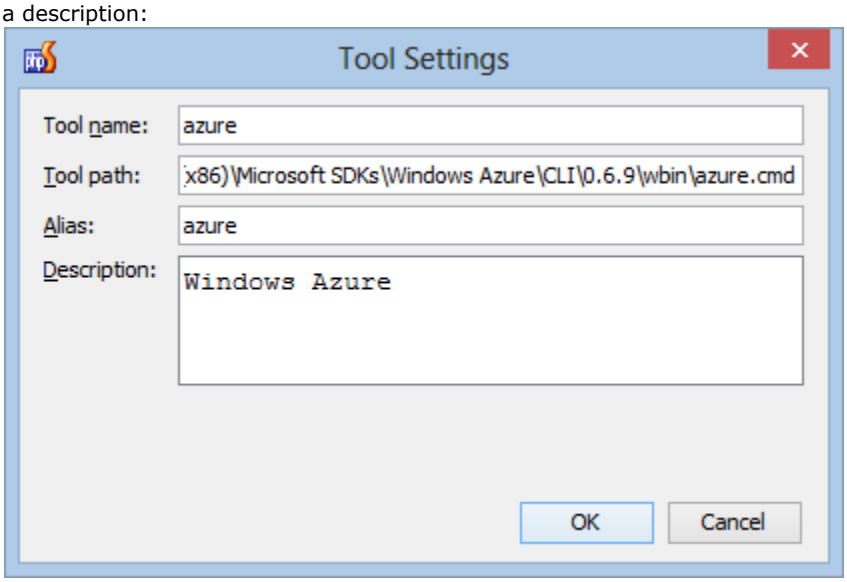
1. From the Project Settings | Command Line Tool Support, add a new tool using the + icon on the toolbar
2. Select one of the supported tools and specify the tool visibility. This can be for the current project only or global (for all projects).
   For this tutorial, we will select Composer as the tool and specify project as the visibility.
3. In the next step, some additional settings will have to be provided. Typically this will be at least the path to the command line tool selected in the previous step.
   For this example, specify the path to composer.phar. If Composer is already being used in the project, PhpStorm should automatically fill out this path.
4. Click OK and note that the tool will be added to the list of command line tools. We can see the alias (which will be used to call the tool) and the full path to the tool. The alias can be changed if we want to.
   In our example, Composer has been added with an alias "c".
5. Close the settings.



Enabling a custom command line tool

To enable a custom tool, a similar workflow can be followed.

1. From the Project Settings | Command Line Tool Support, add a new tool using the + icon on the toolbar
2. Select Custom tool. The tool can be enabled for the current project only or global (for all projects).
3. In the next step, some additional settings will have to be provided: the tool name, the full path to the tool, an alias, and a description:

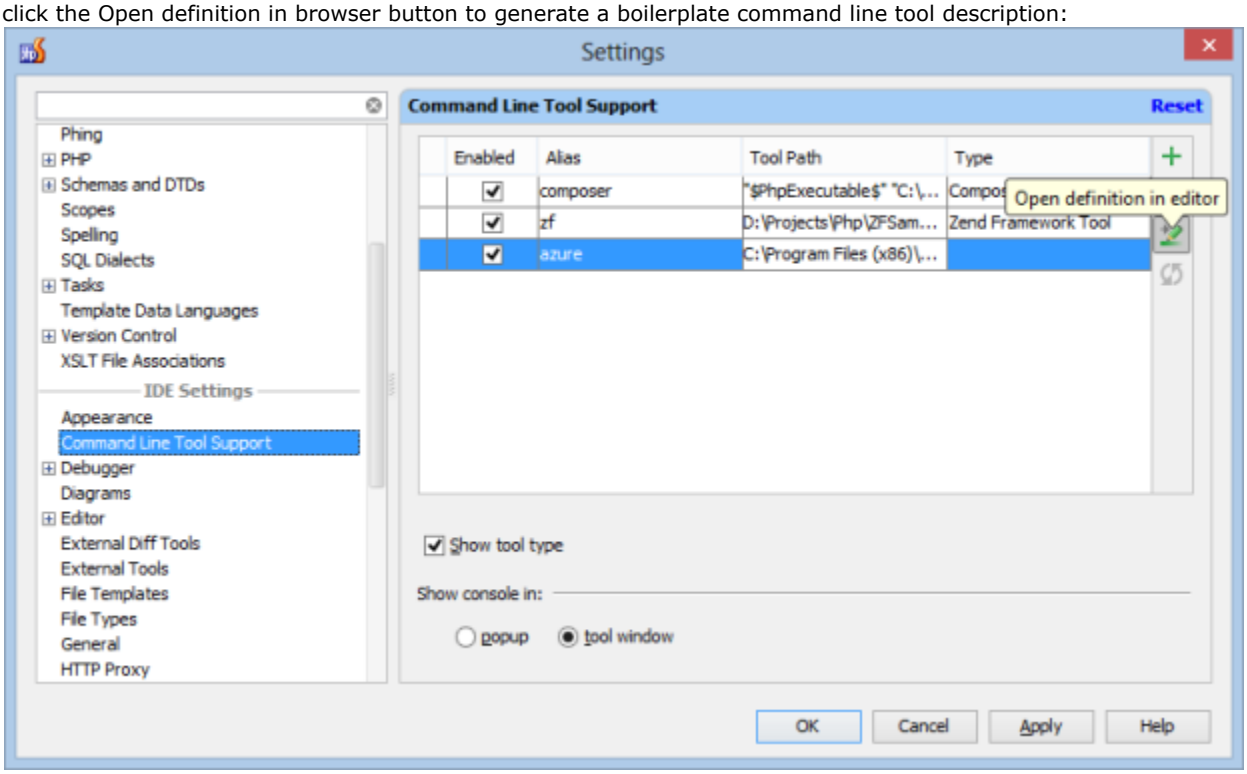


4. Click OK and note that the tool will be added to the list of command line tools. We can see the alias (which will be used to call the tool) and the full path to the tool.
5. Optionally, we can provide the IDE with autocompletion information.
6. Close the settings.

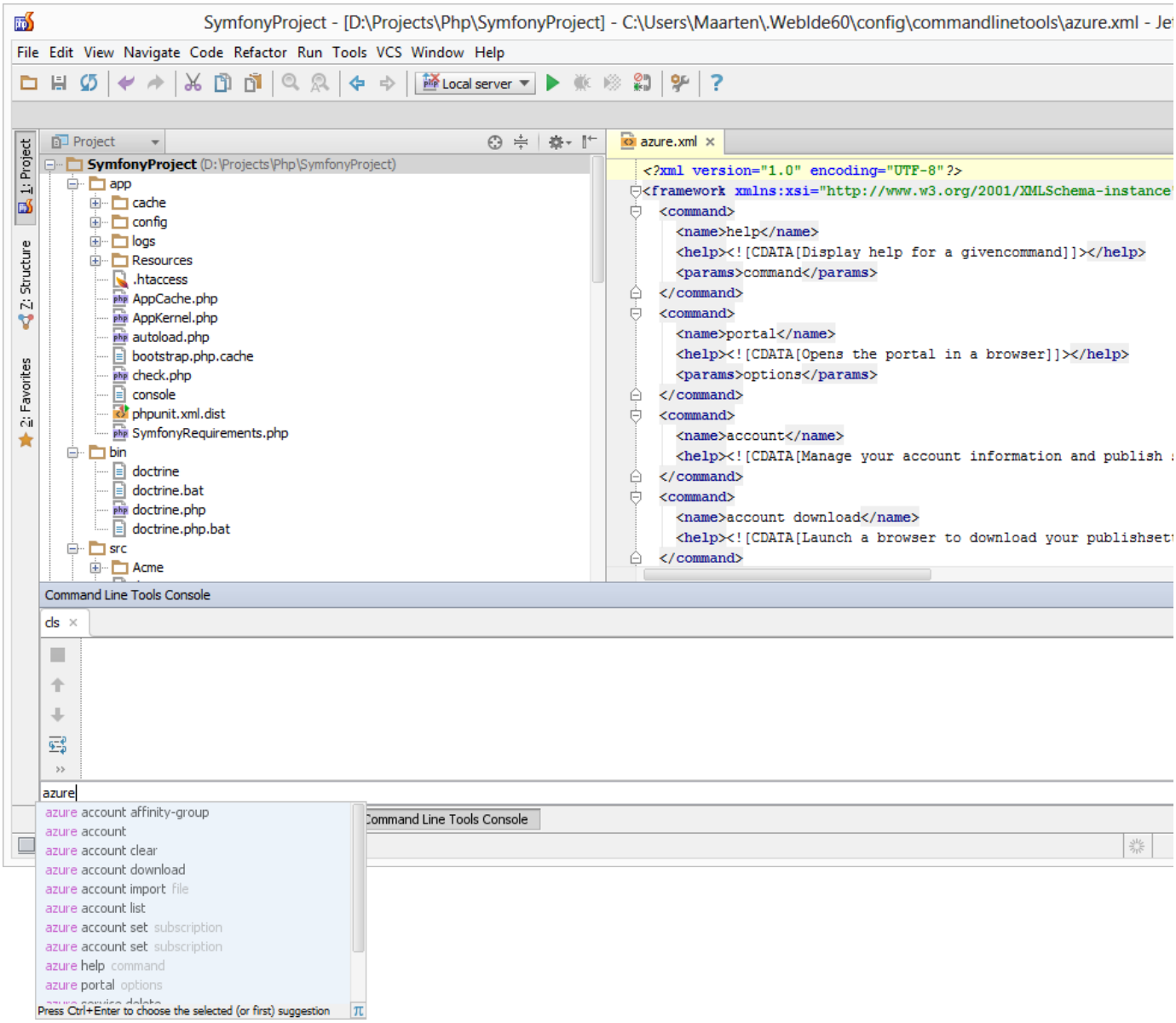
Adding autocompletion for a custom command line tool

PhpStorm can not determine autocompletion for every custom tool we add automatically. Fortunately, we can create our own definition.

1. From the Project Settings | Command Line Tool Support, select the tool to create autocompletion information for and click the Open definition in browser button to generate a boilerplate command line tool description:



2. The editor will open an XML file in which information about the custom tool can be added. We can now define commands, parameters and help information for our custom command line tool. You can also check Command Line Tools - Custom Tools Command Definitions.
3. After saving the XML file containing the description, PhpStorm will recognize the command and provide us with autocompletion for our custom command line tool.



(a Gist for the above screenshot can be found here)

Tweet