Drupal-Specific Coding Assistance

On this page:

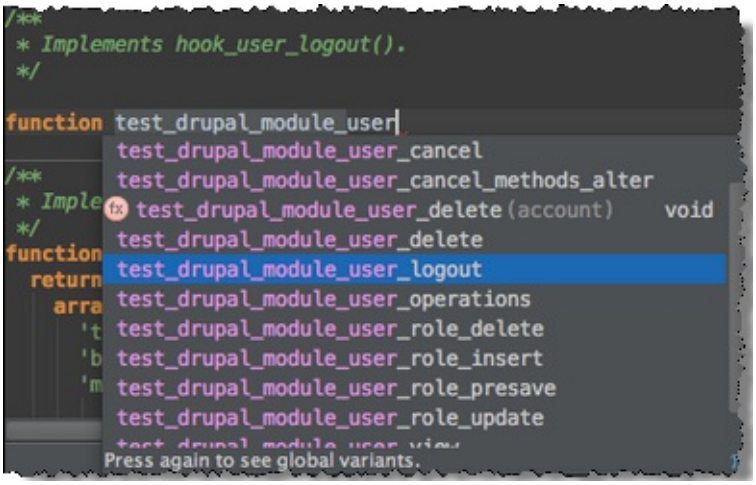
Using Drupal Hooks In IntelliJ IDEA

IntelliJ IDEA provides full native support of Drupal hooks in `.module` files.

Completing hook declarations

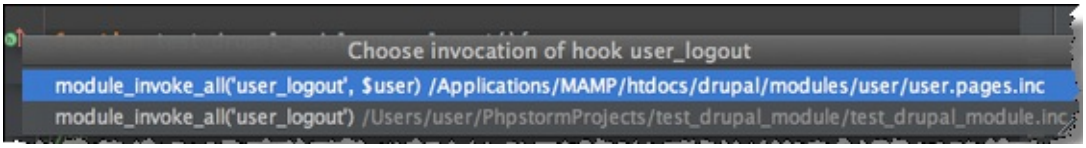
IntelliJ IDEA indexes any hook invocation whereupon hook names become available in code completion for creating hook implementations.

- Start typing a hook name and press **Ctrl+Space**.



Navigating to a hook invocation from the editor

- To navigate to a hook invocation from the editor, click the icon in the gutter.
- In case of multiple invocations, IntelliJ IDEA displays a list of available hook invocations so you can choose which one to navigate to.



You will be navigated to the line where the relevant hook is invoked with `module_invoke_all()`, `module_invoke()`, or `drupal_alter()`.

Quick documentation look-up for hooks

You can view *quick documentation* for a *Drupal* hook in IntelliJ IDEA. Documentation is taken from `.api.php` files provided by `core` and many other modules for reference purposes.

- To view documentation on a hook, position the cursor at the hook name in question and choose **View | Quick Documentation Lookup** or press **Ctrl+Q** or **Alt+Button2 Click**.

Setting the Drupal code style in a project

IntelliJ IDEA automatically offers to apply the pre-configured *Drupal Coding Standards* (code style) if a project is recognized as a *Drupal Module*, or if the *Drupal* integration is enabled in an existing project, or when you create a new project with a a *Drupal Module*. However, you can at any time change or customize this setting on the **Code Style: PHP** page of the **Settings** dialog box.

To have the pre-configured *Drupal code style* applied to a project, do one of the following:

- In the **Event Log** tool window, click the **Set** link next to the `Drupal-style formatting can be set for this project` message.
- Use the Code Style: PHP page of the **Settings** dialog box:
  1. Open the **Settings** dialog box by choosing **File | Settings** on the main menu, click **Code Style**, and then click **PHP**.
  2. On the **Code Style: PHP** page that opens, click the **Set from** link.
  3. On the pop-up menu, choose **Predefined**, then choose **Drupal**.

Checking code against the Drupal coding standards

With IntelliJ IDEA, you can use two tools that detect violations against the Drupal coding standard: Coder and PHP Code Sniffer. They ensure your code remains clean and consistent and help prevent some common semantic errors made by developers.

- To use *PHP Code Sniffer*:
  1. Install *PHP Code Sniffer* as a *PEAR* package or using the *Composer* tool. For detailed installation instructions, see https://github.com/squizlabs/PHP_CodeSniffer.
  2. Register *PHP Code Sniffer* with IntelliJ IDEA and configure it as a IntelliJ IDEA inspection. For details, see Using PHP Code Sniffer Tool.
- To use *Coder*, the native *Drupal* code quality tool:
  1. Download the Drupal Coder module (7x-2.0 version is recommended). You don't need to install or use the *Drupal Coder module*, it contains the *Drupal Coding Standards* inside.
  2. Unpack the downloaded archive and find the `coder_sniffer/Drupal` subdirectory inside. You need to move the contents of this directory to `<php installation folder>/CodeSniffer/Standards/Drupal`. This directory must contain the `ruleset.xml` file and some other subdirectories.

Viewing Drupal API documentation from IntelliJ IDEA

You can reach the *Drupal API Documentation* at https://api.drupal.org/api/drupal right from IntelliJ IDEA.

- Select the symbol you are interested in and choose **Search in Drupal API** on the context menu of the selection.

Using the Drush command line tool

Drush is a command line shell and scripting interface for *Drupal*. With IntelliJ IDEA, you can use *Drush 5.8* and higher.

1. Download and install *Drush* as described at https://github.com/drush-ops/drush.
2. Integrate *Drush* with IntelliJ IDEA by configuring it a s *Command Line Tool*:
    1. Open the project settings and click **Command Line Tool Support**.
    2. On the Command Line Tool Support page, click the **Add** button. In the **Choose Tool to Add** dialog box that opens, choose **Drush** and specify the tool's visibility. The available options are **Project** (available only in the current project) and **Global** (available across all IntelliJ IDEA projects). Click **OK**.
    3. In the **Drush** dialog bo that opes, specify the path to the *Drush* executable file. IntelliJ IDEA automatically fills in the default location, which is usually `C:/ProgramData/Drush/drush.bat` on Windows and `/usr/bin/drush` on Mac OS or Linux. If you followed the standard installation procedure, the predefined path will be correct, just click **OK**, whereupon IntelliJ IDEA loads command definitions automatically and returns to the **Command Line Tool Support** page.
    4. In the **Alias** text box, specify the alias to use in calls of tool commands. Accept the default alias or edit it, if necessary.
    5. To activate the detected command set, select the **Enable** check box.
3. To run a *Drush* command, choose **Tools | Run Command...** on the main menu or press `Ctrl+Shift+X` (`Cmd-Shift-X` on OS X).Ctrl+Shift+X (Cmd-Shift-X on OS X).

For details about configuring and running command line tools in IntelliJ IDEA, see Using Command Line Tools.

See Also

Procedures:

- Using Drupal with IntelliJ IDEA

Reference:

- Drupal
- Drupal Module Dialog

External Links:

- http://confluence.jetbrains.com/display/PhpStorm/Drupal+Development+using+PhpStorm

Web Resources:

- Developer Community