Creating PHPUnit Tests

You can write unit tests manually or have test stubs generated automatically based on the PHP classes that are subject for testing.

A *Test* class is a PHP class with the name `<production class>Test.php`. For example, if the class to test is `Myclass.php`, the tests for it will be in `MyclassTest.php`.

IntelliJ IDEA also generates tests for classes defined among others within a PHP file. If several production classes are defined in one single file, for each generated test class IntelliJ IDEA will create a separate file.

Test classes are inherited from `PHPUnit_Framework_TestCase`.

The data for generating PHP test classes are specified in the **Generate PHPUnit Test** dialog box.

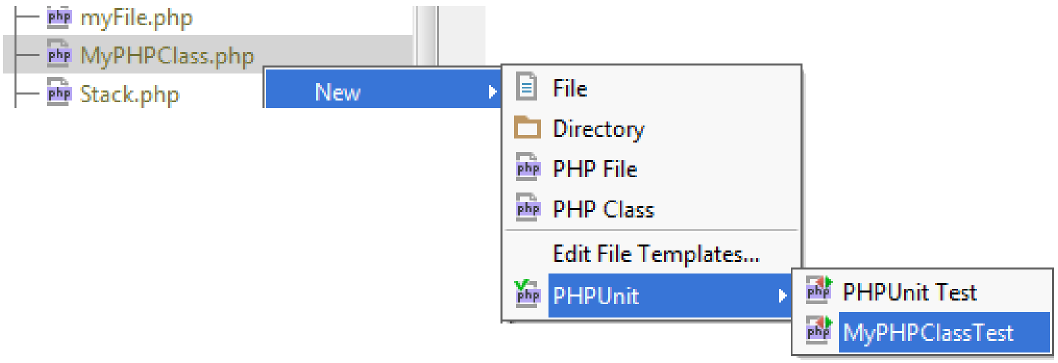
On this page:

Prerequisites

Make sure, the PHPUnit tool is installed on your machine and enabled in IntelliJ IDEA. For more information, see http://www.phpunit.de/manual/current/en/installation.html and Enabling PHPUnit Support.

Generating a test for a PHP class defined in a separate file

1. In the **Project** tool window, select the PHP class to create unit tests for, e.g. `MyPHPClass` as shown in the image below, and choose **New | PHPUnit | PHPUnit Test** on the context menu of the selection.



2. In the **Generate PHPUnit Test** dialog box that opens, specify the following data:
    1. The fully qualified name of the class to be tested in the **Class to test** area. The specified name will be used to propose the **Test Class Name**. To use completion, press **Control+Space** and choose the relevant production class from the pop-up list. By default, the **Name** text box displays the name of the class on which the test generation was invoked.
    2. The *name of the test class*. IntelliJ IDEA automatically composes the name from the production class name as follows: `<production class>Test.php`. The test class name is displayed in the **Name** text box of the **Test Class** area.
    3. The *directory* where the file with the test class will be stored. By default, it is the directory where the production class is stored. The default value is displayed in the **Directory** text box of the **Test Class** area. To specify another folder, click the **Browse** button and choose the relevant folder in the dialog box that opens. To use completion, press **Control+Space** and choose the relevant folder from the pop-up list.
    4. The *namespace* the test class will belong to. IntelliJ IDEA completes the *namespace* automatically based on the *directory* and displays the generated value in the **Namespace** text box. When the *directory* is changed, the *namespace* is changed accordingly. To use completion, press **Control+Space** and choose the relevant namespace from the pop-up list.
    5. The *test file name*. By default, the name is displayed in the **Name** text box and is the same as the test class name.

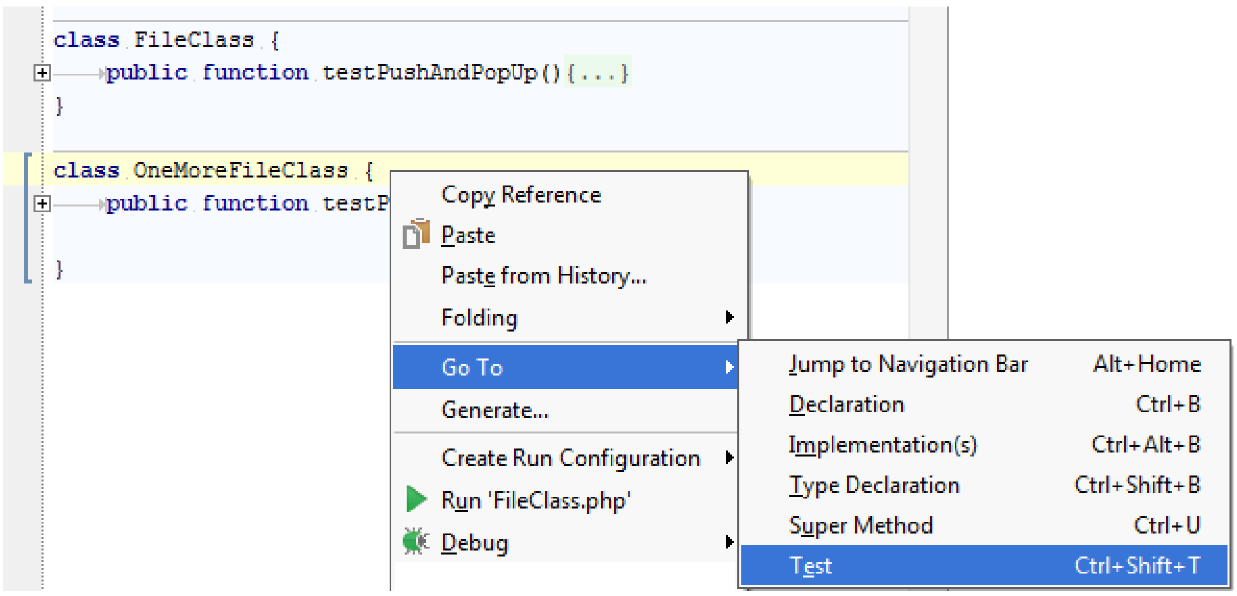
    Check, accept, or update the predefined settings and click **OK** to initiate the test generation.

    Alternatively, you select the PHP production class in question, choose **New | PHPUnit | PHPUnit Test**, and fill in the data manually.

When the test is ready, navigate back to the production class by choosing **Navigate | Go to Test Subject**. For details, see Navigating Between Test and Test Subject.

Generating a test for a PHP class defined among others within a PHP file

1. In the editor, open the file with the class to generate the test for.
2. Do one of the following:
   - On the main menu, choose **File | New | PHPUnit | PHPUnit Test**.
   - On the context menu, choose **Go To | Test** on the context menu, then choose **Create New Test** in the pop-up list.



```
class FileClass {
    public function testPushAndPopUp(){...}
}

class OneMoreFileClass {
    public function testP

}
```

3. In the **Generate PHPUnit Test** dialog box that opens, specify the following data:
   1. The fully qualified name of the class to be tested in the **Class to test** area. To use completion, press **Control+Space** and choose the relevant production class from the pop-up list. The specified name will be used to propose the **Test Class Name**.
   2. The *name of the test class*. IntelliJ IDEA automatically generates the name and displays it in the **Name** text box of the **Test Class** area.
   3. The *directory* where the file with the test class will be stored. By default, it is the directory where the production class is stored. The default value is displayed in the **Directory** text box of the **Test Class** area. To specify another folder, click the **Browse** button and choose the relevant folder in the dialog box that opens. To use completion, press **Control+Space** and choose the relevant folder from the pop-up list.
   4. The *namespace* the test class will belong to. IntelliJ IDEA completes the *namespace* automatically based on the *directory* and displays the generated value in the **Namespace** text box. When the *directory* is changed, the *namespace* is changed accordingly. To use completion, press **Control+Space** and choose the relevant namespace from the pop-up list.
   5. The *test file name*. By default, the name is displayed in the **Name** text box and is the same as the test class name.

   Check, accept, or update the predefined settings and click **OK** to initiate the test generation.

Generating a PHPUnit test method

Besides generating test class stubs, IntelliJ IDEA can help you populate them by generating stubs for test methods.

1. Open the required test class in the editor, position the cursor anywhere inside the class definition, and do one of the following:
   - On the main menu, choose **Code | Generate**, then choose **PHPUnit Test Method** from the **Generate** pop-up list.
   - On the context menu, choose **Generate**, then choose **PHPUnit Test Method** from the **Generate** pop-up list.
2. Set up the test *fixture*, that is, have IntelliJ IDEA generate stubs for the code that emulates the required environment before test start and returns the original environment after the test is over. For more details, see http://phpunit.de/manual/3.7/en/fixtures.html.
   1. On the context menu, choose **Generate | Override method**.
   2. From the **Choose methods to override** dialog box that opens, choose **SetUp** or **TearDown** and click **OK**.

See Also

Procedures:

- Testing PHP Applications

Reference:

- Create New PHPUnit Test

External Links:

- http://confluence.jetbrains.com/display/PhpStorm/Creating+PHPUnit+Tests+in+PhpStorm

Web Resources:

- Developer Community