



PhpStorm Workshop

Maarten Balliauw
Technical Evangelist

maarten.balliauw@jetbrains.com

Before we start

Make sure you have PhpStorm 2016.1+ installed
either full or free trial from <http://jetbrains.com/phpstorm>

Make sure you have a Vagrant 1.8.*+ and VirtualBox 5.*+

Make sure you have git command line

How it all works

This course covers PhpStorm 2016.1

But most is applicable to earlier versions

There will be some theory

And some practice

We will not cover every knob and bolt

Do ask questions!

Keyboard shortcuts

PhpStorm features multiple keymaps

Defaults per platform   

Other IDEA-based IDE's use a similar keymap

Cheat sheet: <http://bit.ly/phpstorm-shortcuts>

Agenda

Look at the IDE

Navigation

Editing

Inspections

Live Templates

Refactoring

Debugging

Todo Explorer

Unit Testing

Version Control

Databases

Deployment

Tools

Plugins

Look at the IDE

Newly installed IDE

Welcome screen

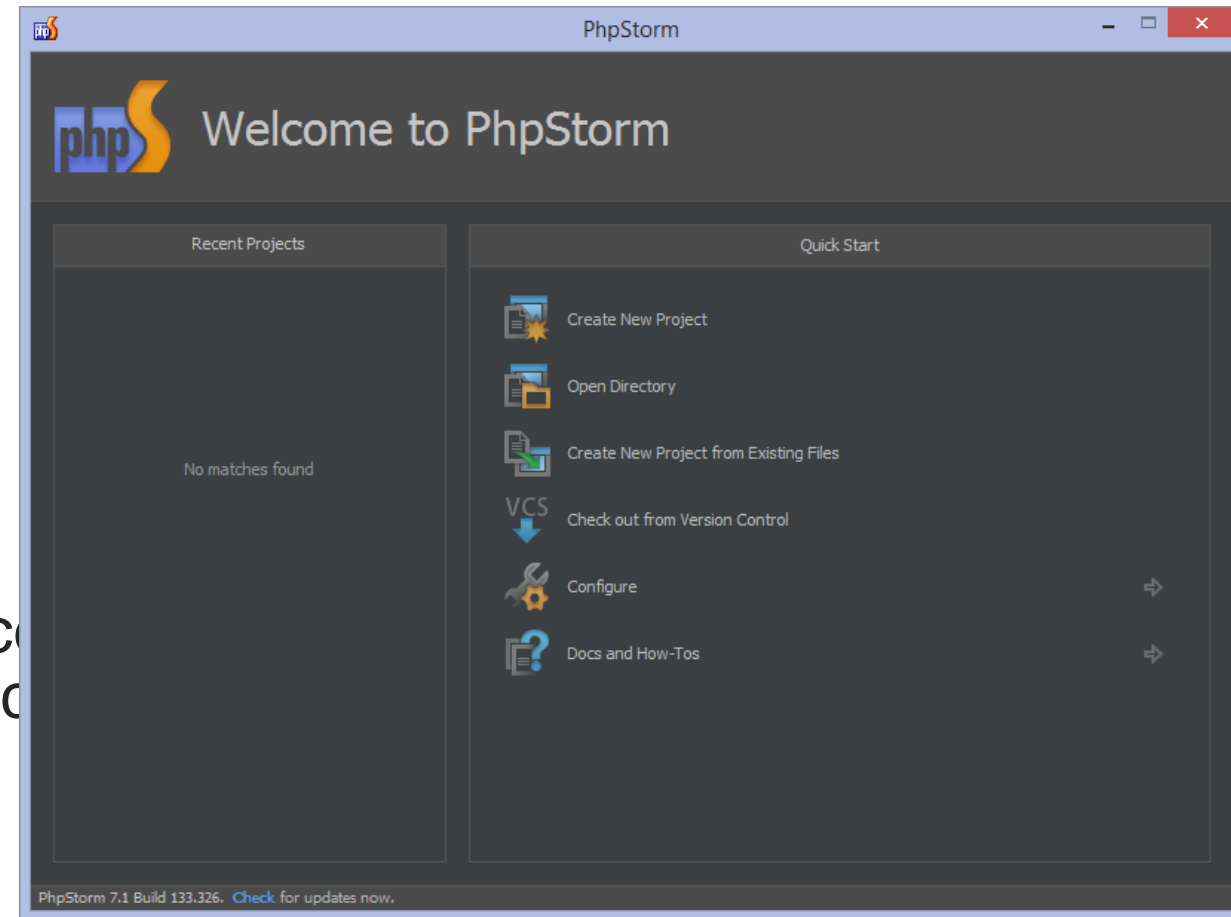
Create project

Open project

Checkout from version control

Configuration

Project? The basis for coding assistance
bulk refactoring, coding style consistency
etc.



Opened project

Menus and toolbars

Execute various commands

Navigation bar

Navigate through project

Status bar

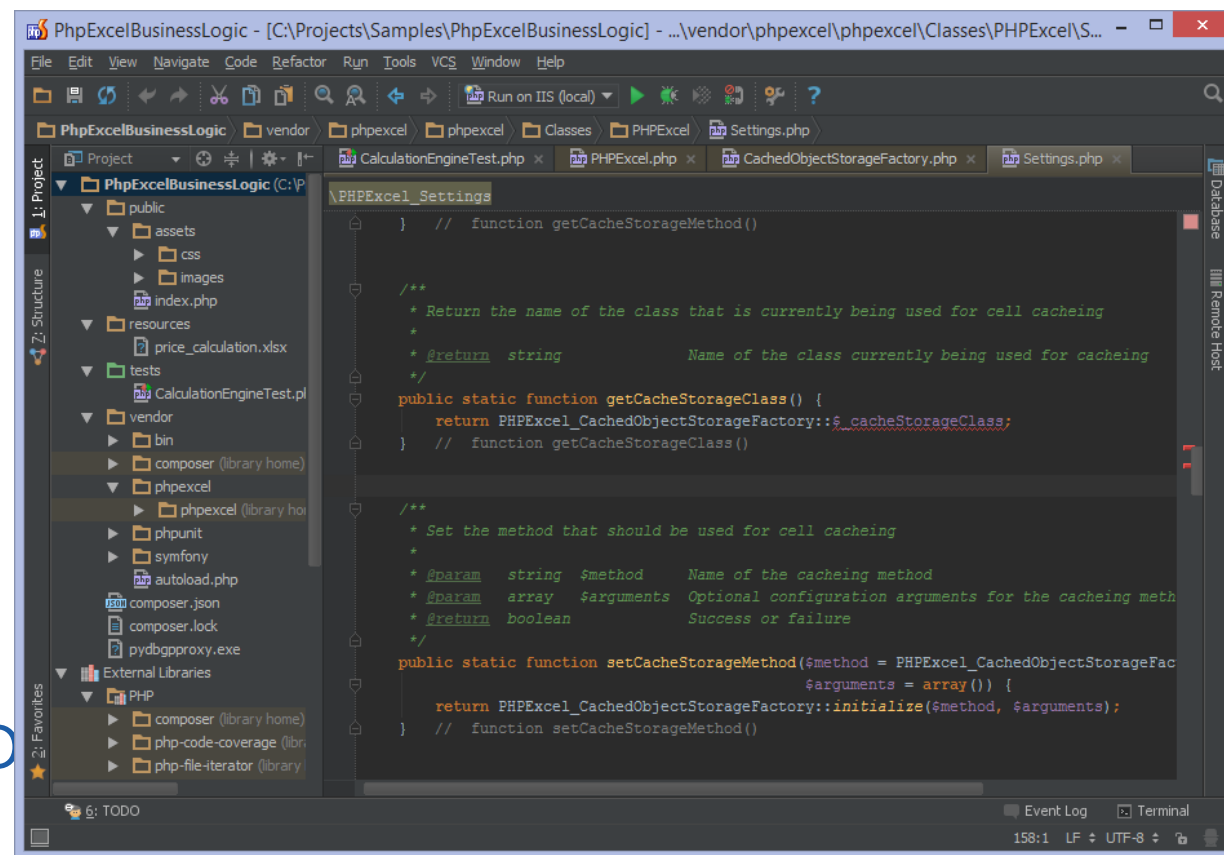
Information about IDE and project

Editor

Where coding happens

Tool windows

Numerous helpers (e.g. database, TODO)



Configuring PHP

Under Project Settings | PHP,
set PHP options.

Language level

Interpreter

Include paths

Debugging

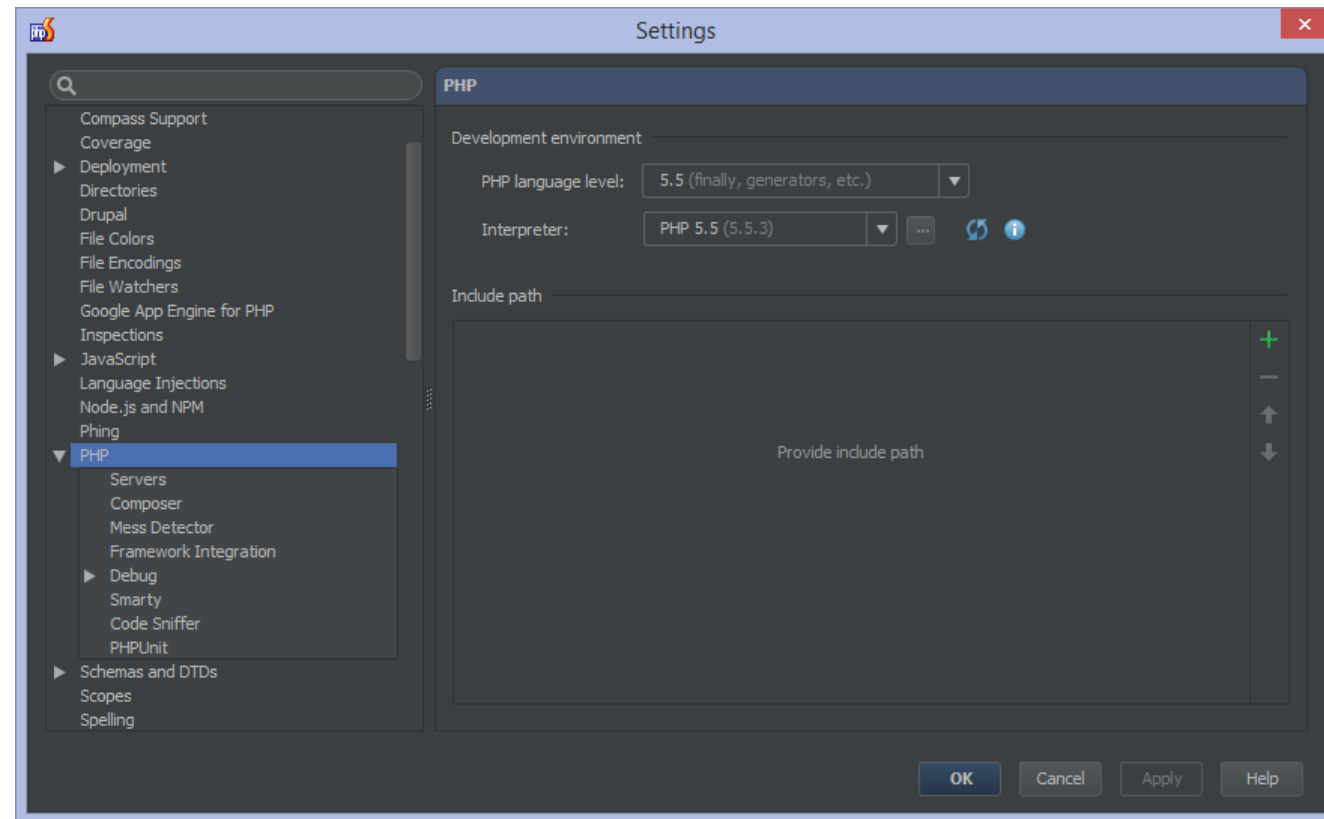
...



Ctrl+Alt+S



Command+,



Configuring PhpStorm

Project settings

Specific for current project

IDE settings

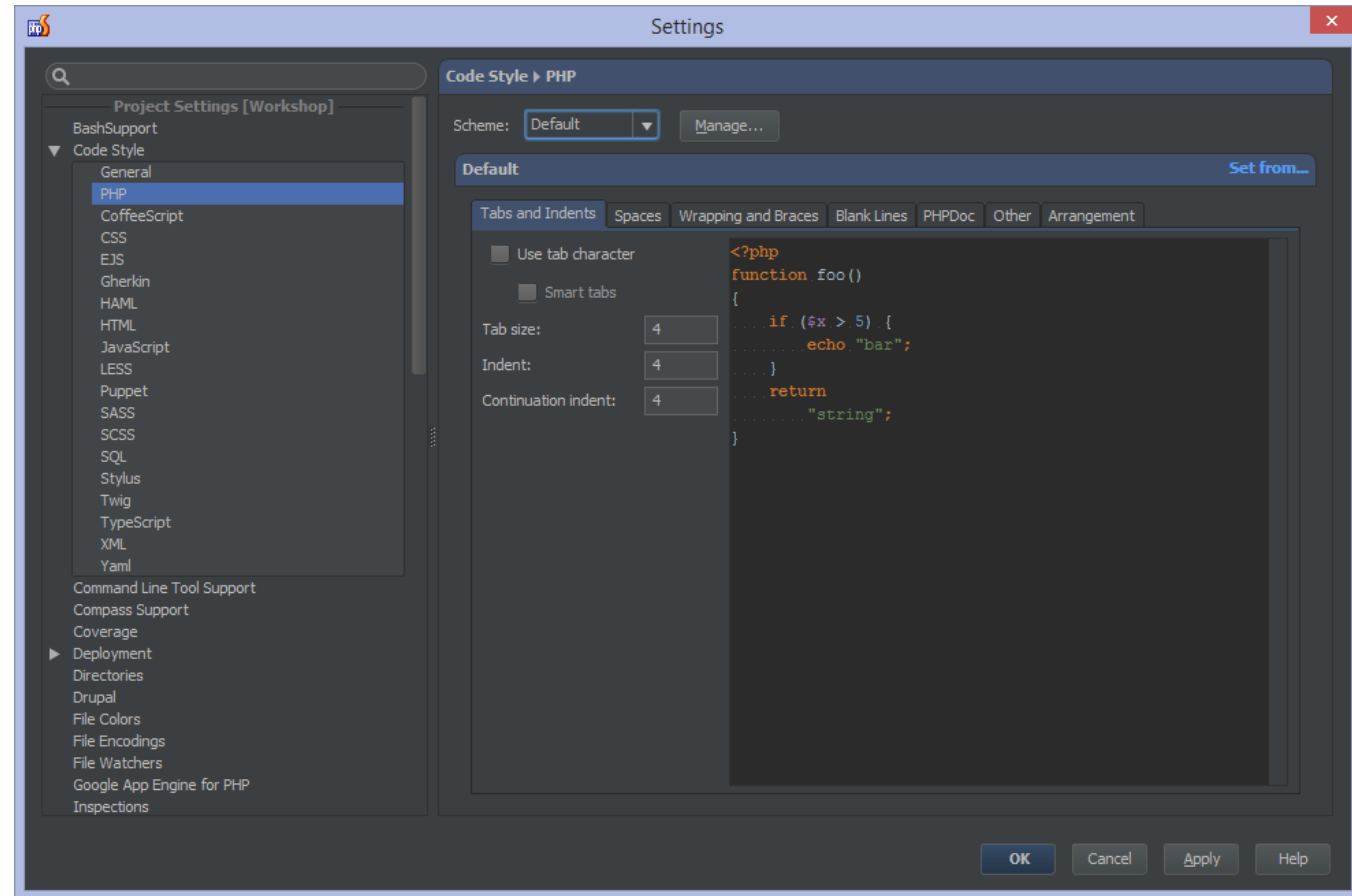
Global for all projects



Ctrl+Alt+S



Command+,



Tabs

Open a document each

Navigate between tabs

Pin tab

Split editor

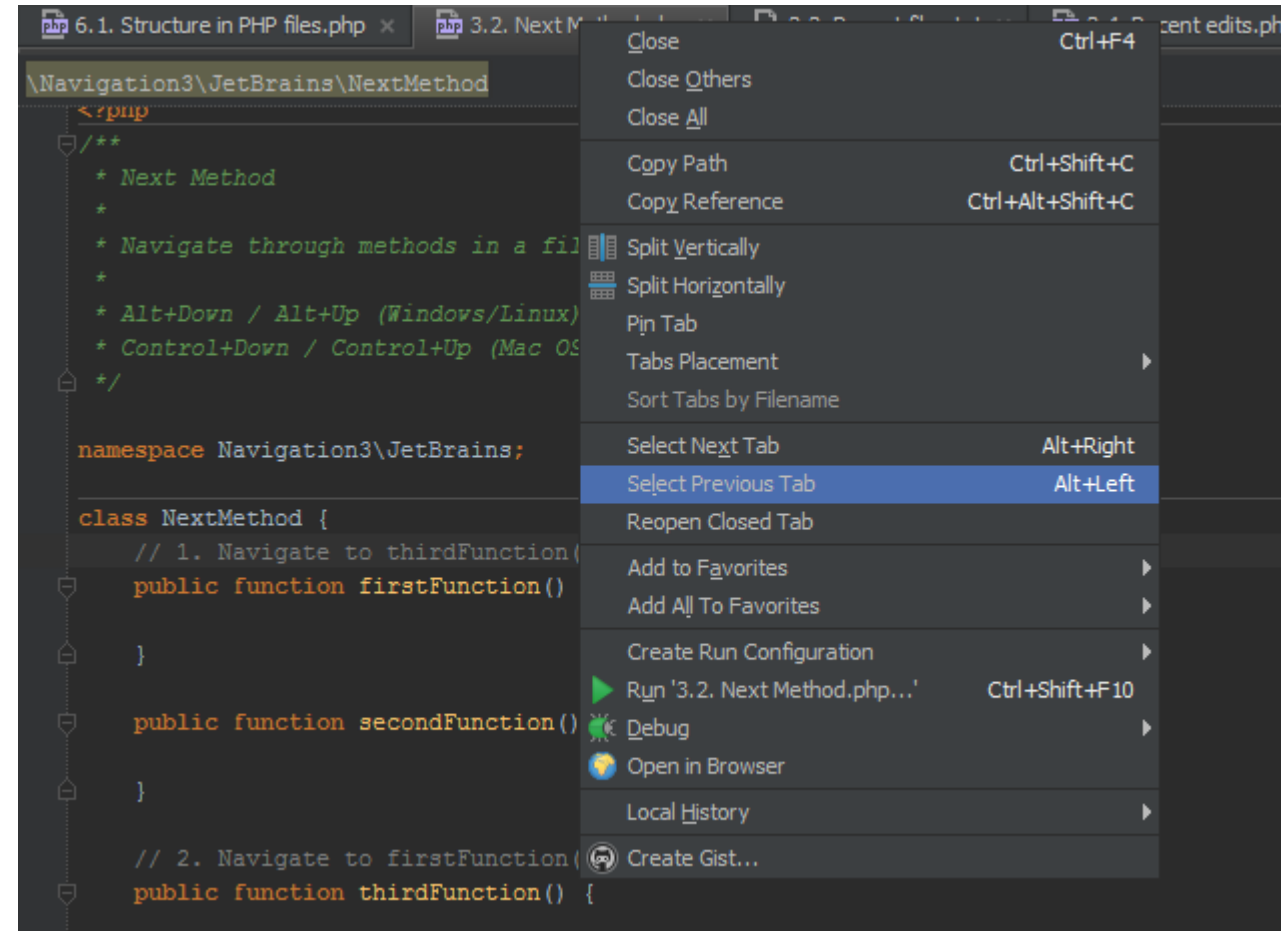
Tabs can be favorited



Alt+Left / Right



Command+Shift+[/] (or Ctrl+Left / Right)



Let's fetch the workshop materials

Start PhpStorm and if a project is opened, close it

Create a new project, name it “Workshop” and select the “Composer project” type

Find the “jetbrains/phpstorm-workshop” package and click OK

Alternatively, use “Checkout from Version Control”

Repository: <https://github.com/jetbrains/phpstorm-workshop.git>

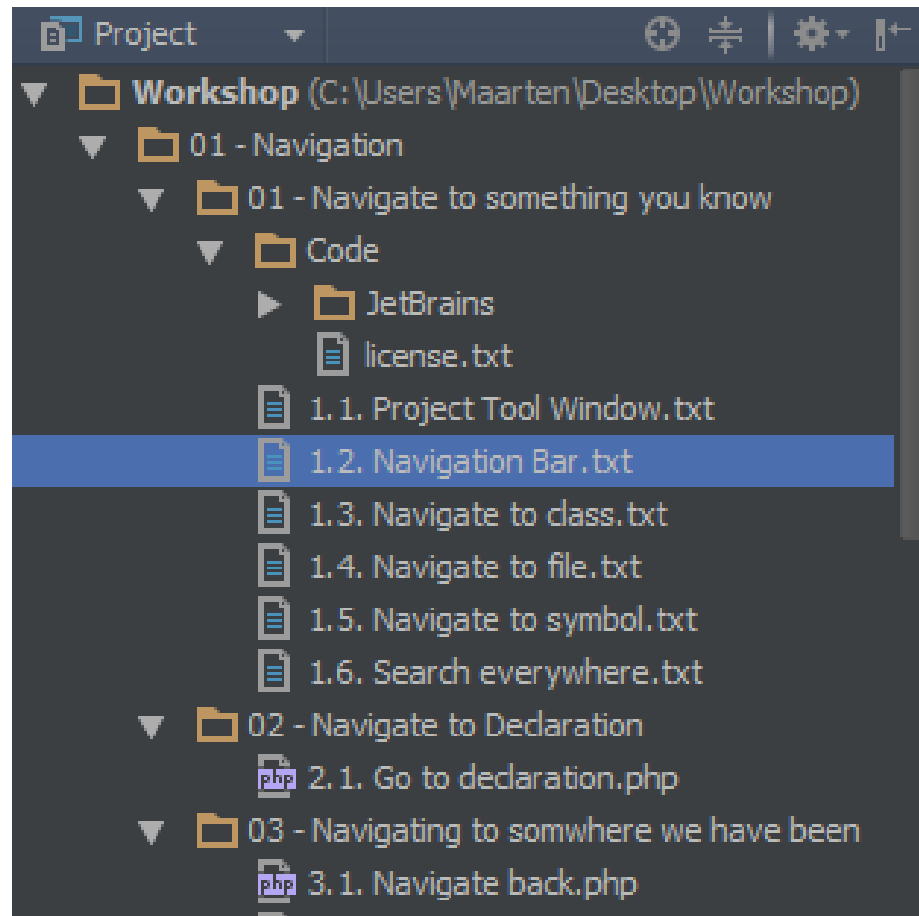
Project / directory name: Workshop



Navigation

Project Tool Window

Shows all files in project.

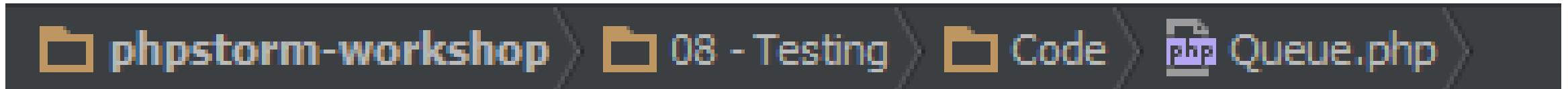


Navigation Bar

Navigate through files in project.

Left/Right navigates path.

Up/Down opens other node in path.



Alt+Home



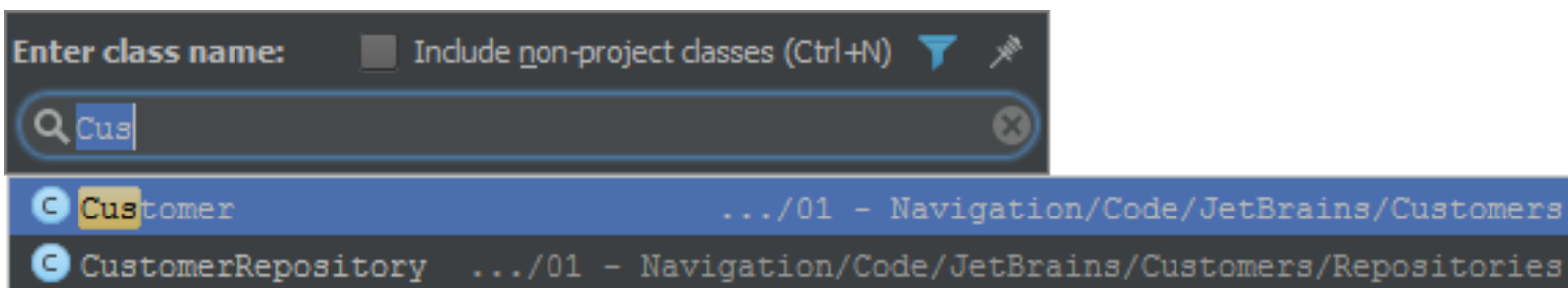
Command+Up

Navigate to Class

Navigates to a given class

can be in PHP or any other
language supported by PhpStorm

use CamelHumps / wildcards



Ctrl+N

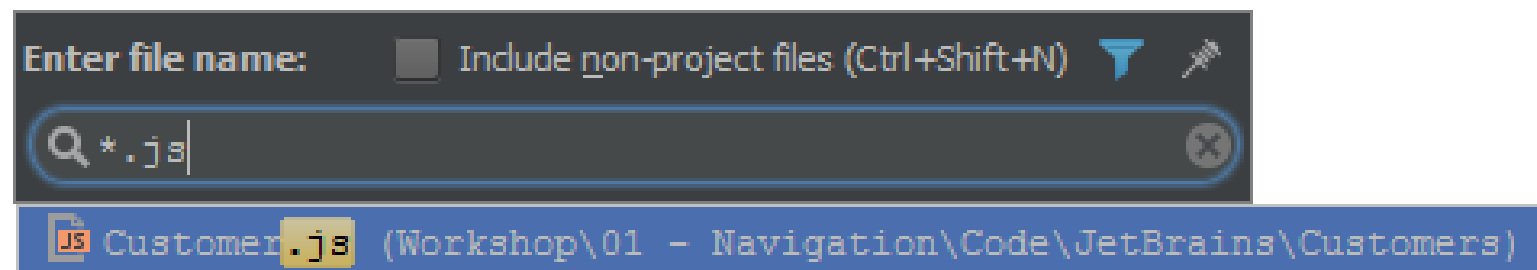


Command+O

Navigate to File

Navigates to a given file
any file type

use CamelHumps / wildcards



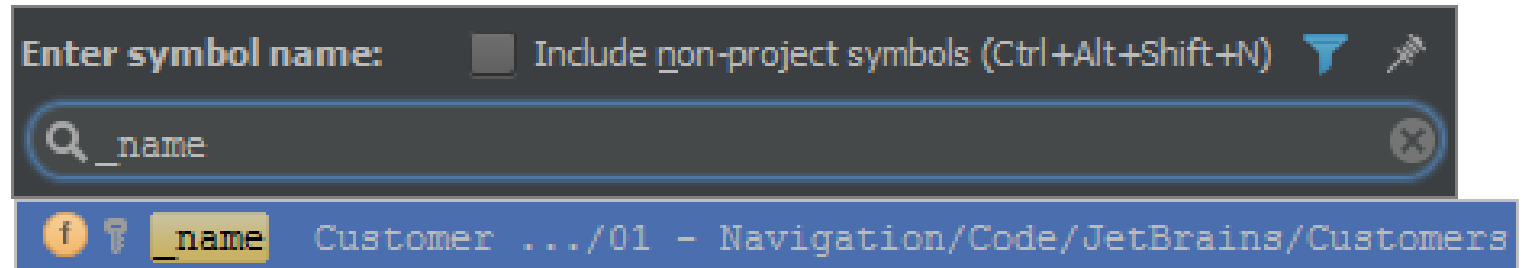
  Ctrl+Shift+N




 Shift+Command+O

Navigate to Symbol

Navigates to a given symbol
can be in PHP or any other
language supported by PhpStorm

use CamelHumps / wildcards



  Ctrl+Shift+Alt+N
 Alt+Command+O

Search Everywhere

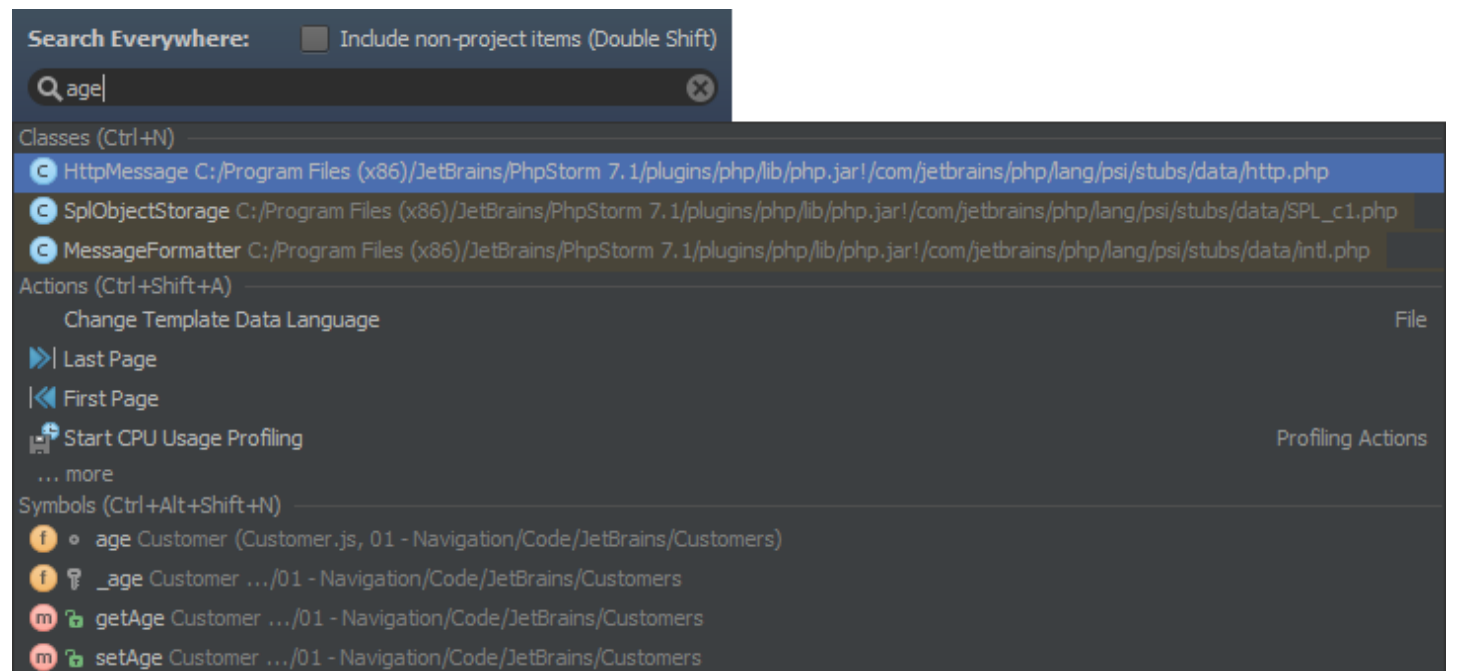
Search classes, file, symbols
and menu actions all at once
use CamelHumps / wildcards



Double shift



Double shift



Navigate to Declaration

Ctrl-click (or Command+Click)
on hyperlink
or use shortcut

```
// 1. Ctrl+Click on the Customer symbol  
function Customer::construct($name, $age) : Customer  
// 3. Go to Declaration for the $age variable passed  
$person = new Customer($name, $age);
```

```
/**  
 * @return int  
 */  
public class Customer {  
    {  
        return $this->_age;  
    }  
}
```



Ctrl+Click or Ctrl+B



Command+Click or Command+B

Navigate Back / Forward

Use the keyboard shortcut



Ctrl+Alt+Left / Right

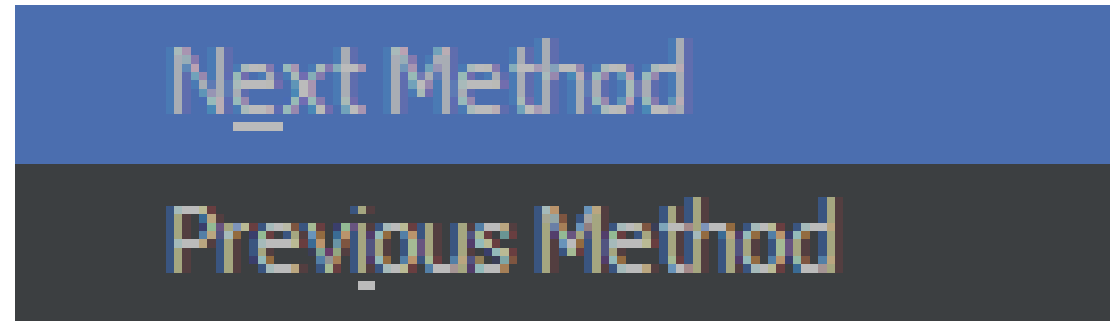


Alt+Command+Left / Right

Next / Previous Method

Use the keyboard shortcut to navigate through methods in a file/class.

Navigates between tags in HTML.
Also works in other file types.



Alt+Down / Alt+Up



Control+Down / Control+Up

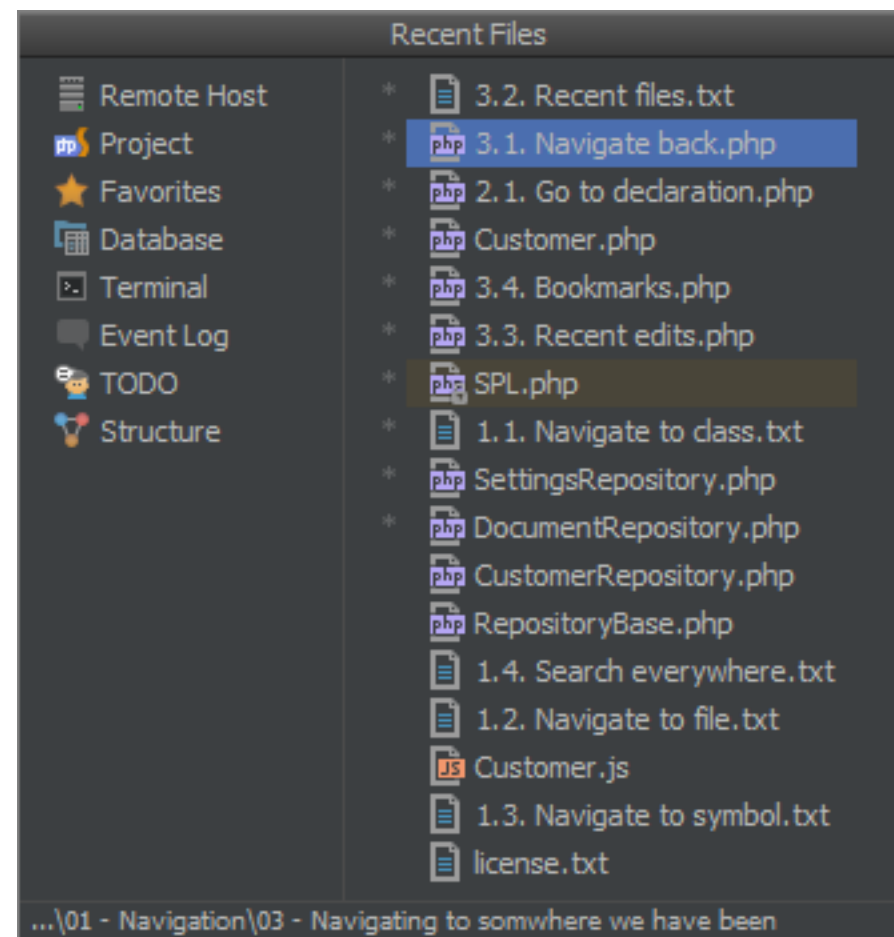
Recent Files

List of recent files

use CamelHumps / wildcards

  Ctrl+E

 Command+E



Navigate to Last Edit Location

Use the keyboard shortcut to navigate back to the last edit location.

Last Edit Location



Ctrl+Shift+Backspace



Shift+Command+Delete

Bookmarks

Navigate between “bookmarked” locations.



F11 Toggle Bookmark

Ctrl+F11 Toggle Numbered Bookmark

Shift+F11 Show bookmarks

Ctrl+0..9 Navigate to numbered bookmark

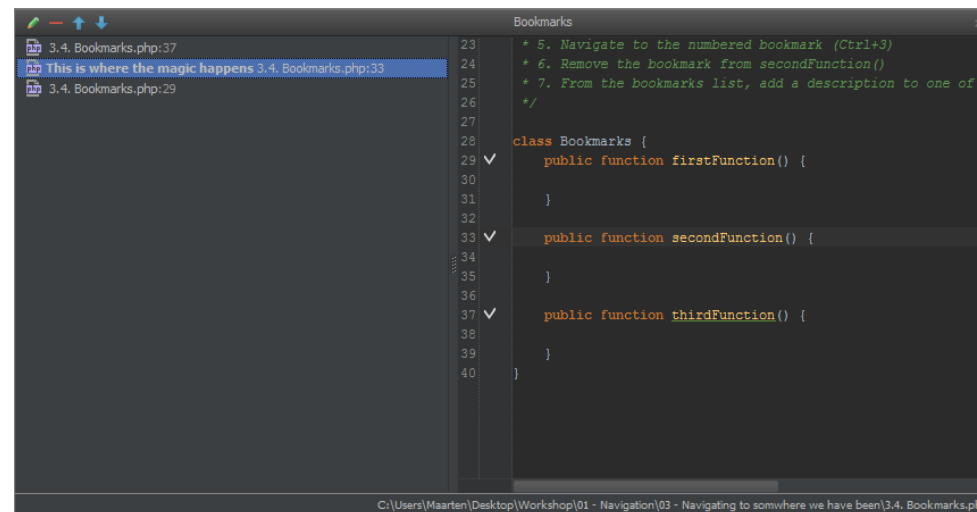


F3 Toggle Bookmark

Alt+F3 Toggle Numbered Bookmark

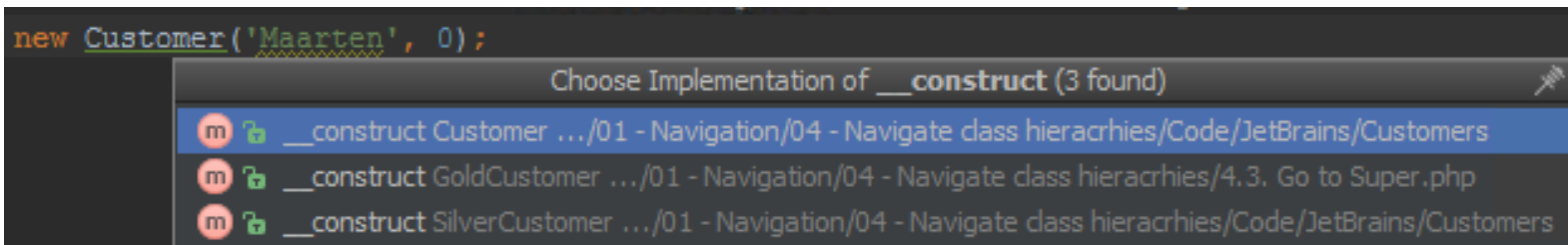
Command+F3 Show bookmarks

Ctrl+0..9 Navigate to numbered bookmark



Go to Implementation

Navigates to the implementation of a given class.



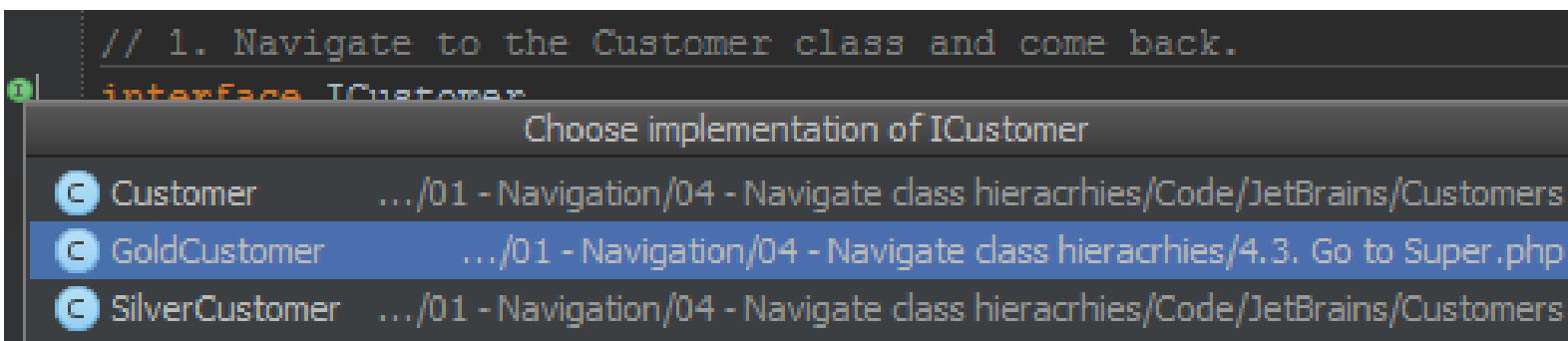
Ctrl+Alt+B



Alt+Command+B

Go to Derived

Navigates to the implementation of a given interface or to a subclass.



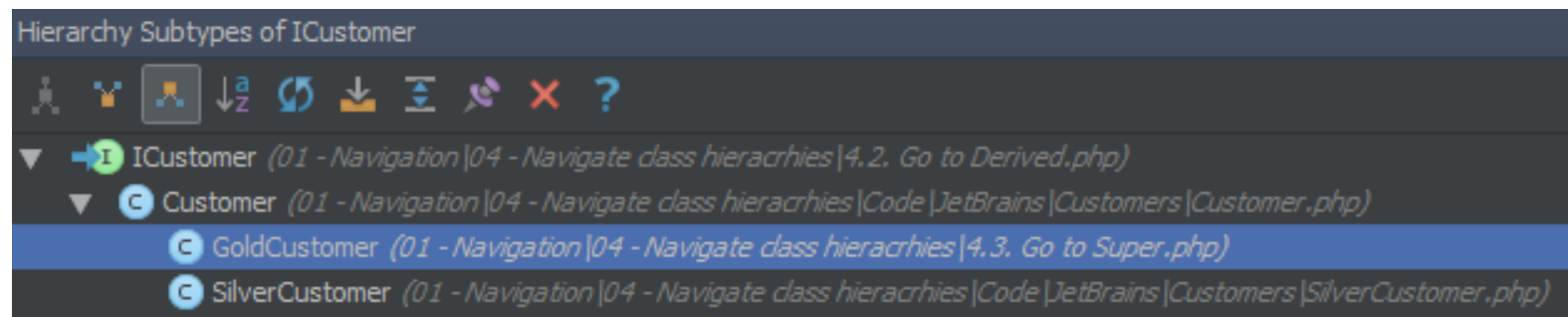
Use the left gutter



Use the left gutter

Go to Super class/method

Navigates to the super class or method.



Ctrl+U (Ctrl+H for hierarchy)



Command+U (Ctrl+H for hierarchy)

Highlight Usages in File

Highlights usages of a symbol in the current file. Different colors for read/write.

Esc to clear highlighting.



Ctrl+Shift+F7

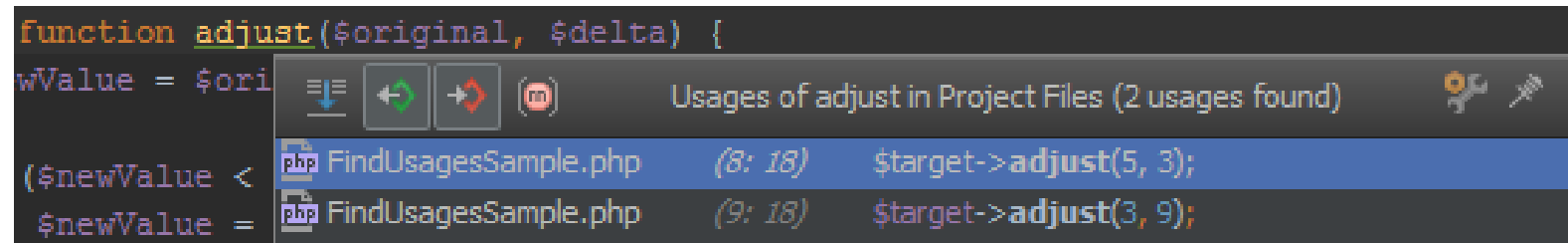


Shift+Command+F7

```
public function adjust($original, $delta) {  
    $newValue = $original + $delta;  
  
    if ($newValue < self::MIN_VALUE) {  
        $newValue = self::MAX_VALUE;  
    }  
  
    if ($newValue > self::MAX_VALUE) {  
        $newValue = self::MIN_VALUE;  
    }  
  
    return $newValue;  
}
```

Find Usages

Find usages of a symbol in the current project. Use tool window or popup.



Alt+F7 (Ctrl+Alt+F7 for popup)



Alt+F7 (Alt+Command+F7 for popup)

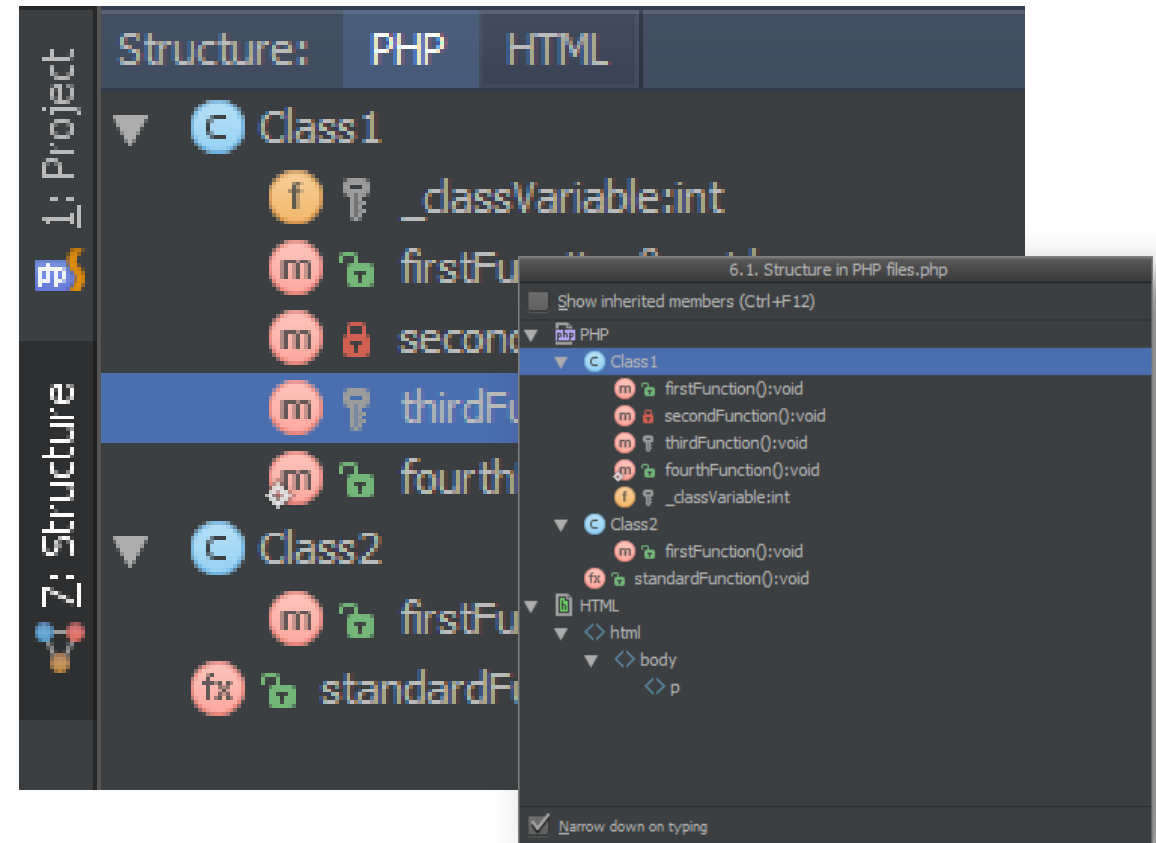
File Structure Tool Window

Displays outline of file: classes, functions (with icons displaying accessibility).

Also shows HTML, JavaScript, CSS, ...

  Alt+7 (Ctrl+F12 for popup)

 Command+7 (Command+F12 for popup)



Editing

Basic Completion

Basic code completion for the name of any class, method or variable.

Hit Ctrl+Space twice for more variants

Also allows path completion

```
public function basicCompletion() {  
    $this->  
    // f Age int  
    // m basicCompletion () void  
    // f Name string  
    // m run () void  
}
```

Invoke completion again to see more variants. >>

  Ctrl+Space (Ctrl+Shift+Enter to complete statement)

 Ctrl+Space (Shift+Command+Enter to complete statement)

Import Completion

Basic code completion when using classes or functions from other namespaces.
Automatically adds import when selected.

```
class ImportCompletion
    implements ICu {
}

ICustomer (Navigation4\JetBrains\Customers) Navigati
Press Ctrl+Period to choose the selected (or first) suggestion and insert a dot afterwards >>
```



Ctrl+Space (Ctrl+Shift+Enter to complete statement)



Ctrl+Space (Shift+Command+Enter to complete statement)

Selecting Code

Various ways of selecting code.



Ctrl+W to expand, Ctrl+Shift+W to shrink



Alt+Up / Down

```
public function extendAndShrinkSelection() {  
    $importantValue = 32;  
    if ($importantValue > 42) {  
        try {  
            echo 'More important than 42?';  
        } catch (Exception $ex) {  
            // 3. Place the cursor on the $ex variable.  
            echo $ex->getMessage();  
        }  
    }  
}
```

Column Selection

Toggle column selection. Allows editing multiple lines in one go.



Alt+Shift+Insert



Shift+Command+8

```
public function columnSelection() {  
    $fo = 3;  
    $fo = 6;  
    $fo = 9;  
    $fo = 'a';  
    $fo = 'b';  
    $fo = 'c';  
}
```

Moving Code

Move code or entire statement up/down.

```
if (3 == 3) {  
    // ...  
    if (1 == 1) {  
        // ...  
        echo 'Number one';  
    }  
    echo 'Number three';  
}
```



Shift+Alt+Up / Down (or Shift+Ctrl+Up / Down for statement)



Shift+Alt+Up / Down (or Shift+Command+Up / Down for statement)

Surround With

Wraps selected text with new content, e.g. try/catch or if statement.



Ctrl+Alt+T and Alt+Enter



Alt+Command+T and Alt+Enter

```
public function moveLine() {  
    echo 'This may be throwing an Exception. Riiiiight!';  
}
```

Surround With

- 1. {}
- 2. if
- 3. if / else
- 4. while
- 5. do / while
- 6. for
- 7. foreach
- 8. try / catch
- 9. function
- 0. <editor-fold...> Comments
- A. region...endregion Comments

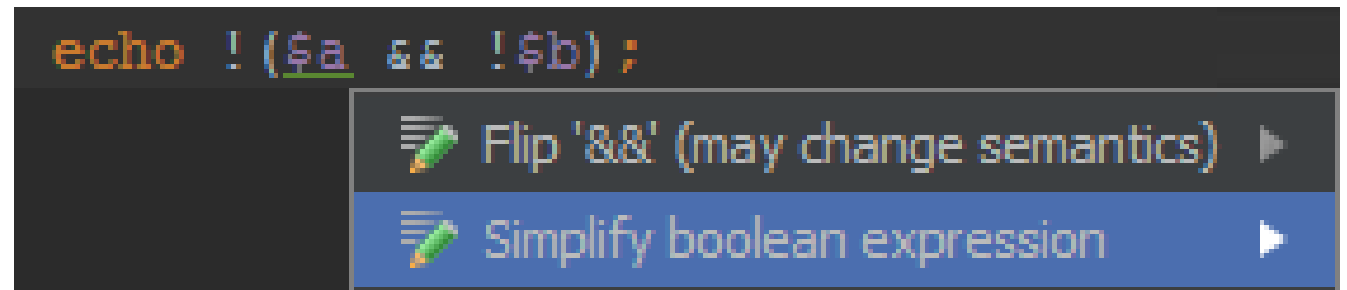
Live templates

- trycatch. Try/catch block

Configure Live Templates...

Intentions

Let the IDE figure out possible actions from context and execute them.



Generate Code

Generate code, e.g. class members, constructor, docblock comments, fields, ...

When used in navigation bar / project tool window, generates a new file.

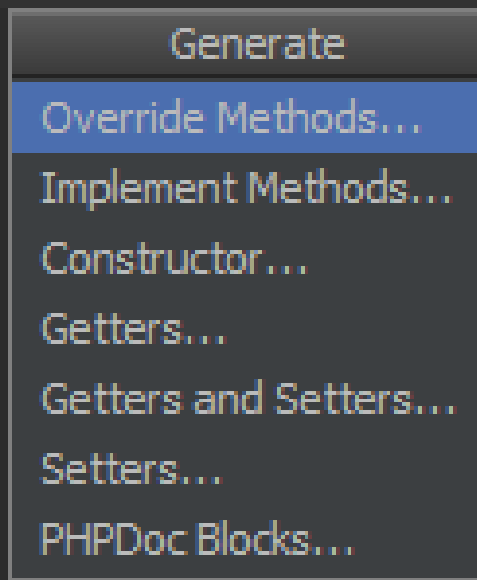


Alt+Insert



Command+N

```
class Person {  
    protected $_firstName;  
    protected $_lastName;  
}
```

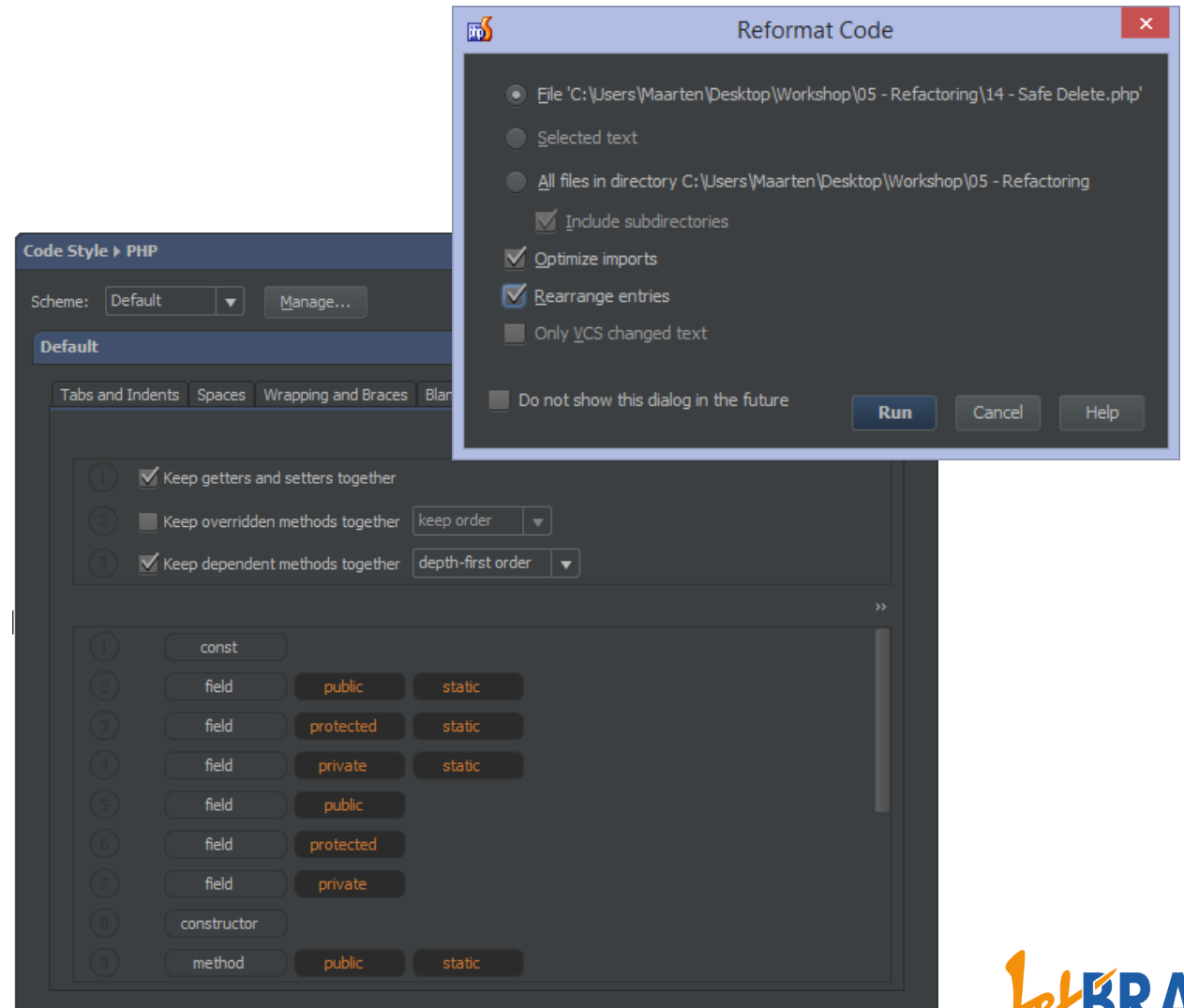


Rearrange/Reformat Code

Rearranges entries in code according to settings. Reformats all code according to code style.

  Ctrl+Alt+L (reformat)

 Alt+Command+L (reformat)



Inspections

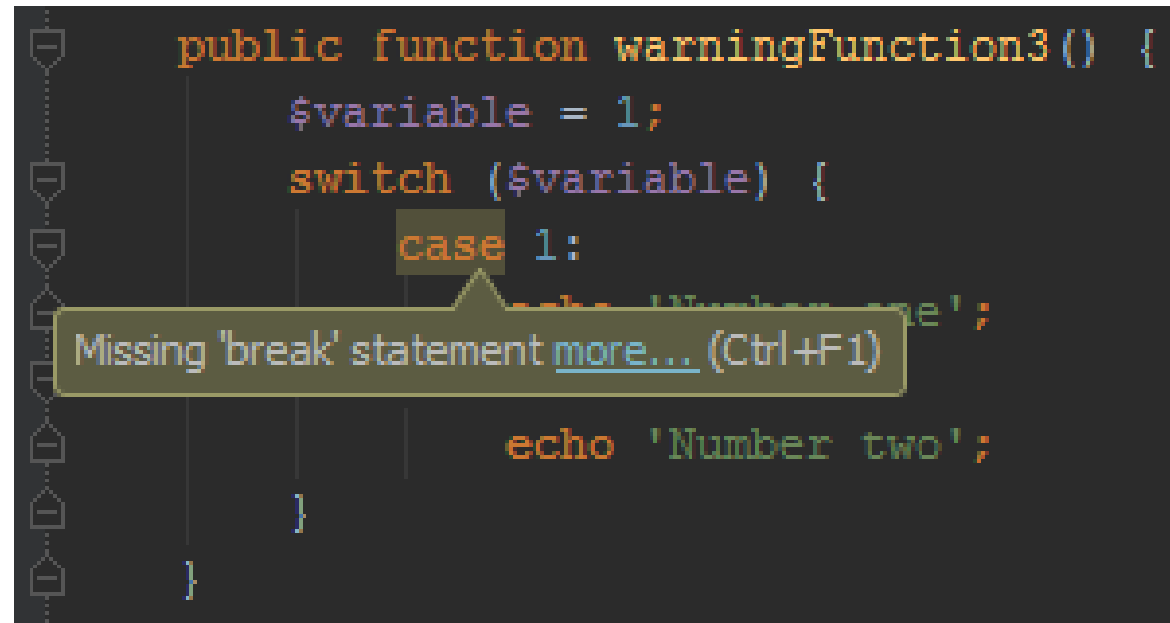
Highlights

In-editor highlighting of code issues.

- Errors
- Warnings
- Suggestions
- Hints
- Dead Code

Can be suppressed.

Support for PHP Code Sniffer /
PHP Mess Detector.



Quick Fixes

Highlights provide a Quick Fix action to help remove the warning.

```
$person = new Person();
```

A screenshot of an IDE showing a code snippet with a warning. The code is `$person = new Person();`. The variable `$person` is underlined with a wavy line, indicating a warning. A Quick Fix menu is open, showing the option 'Import Class' with a lightbulb icon and a right arrow.

Navigation

Navigate back and forth between code issues.

Next Highlighted Error

Previous Highlighted Error



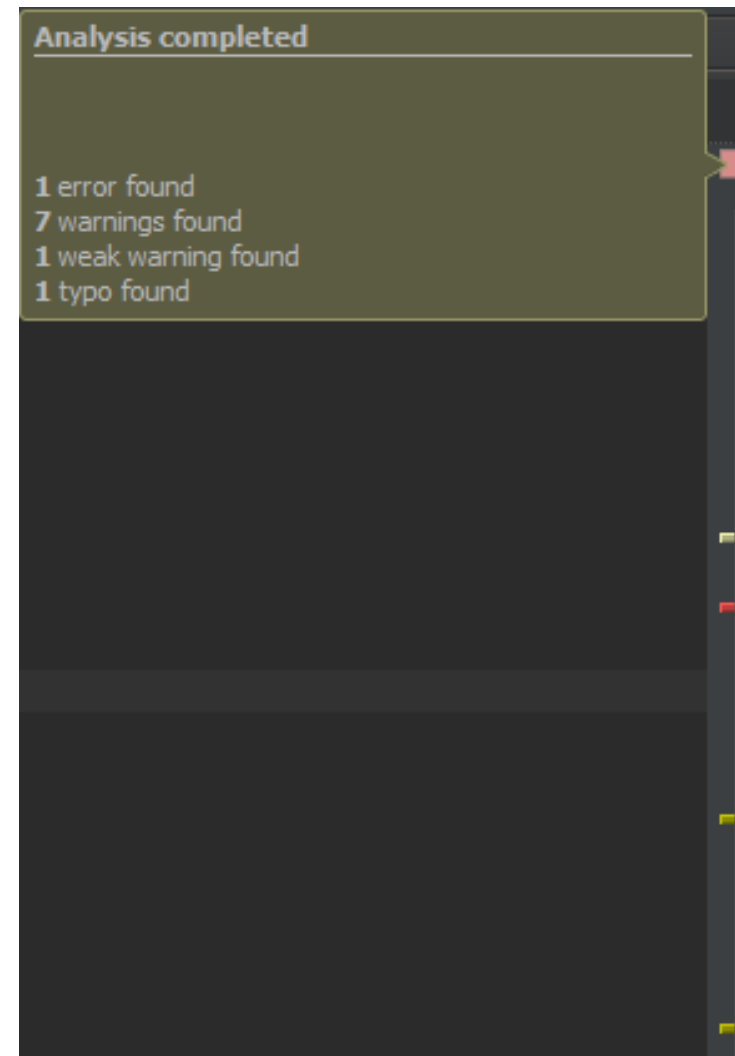
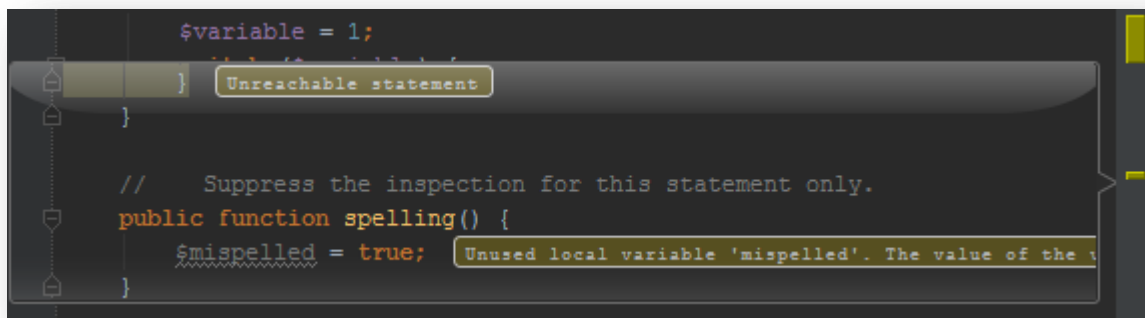
F2 / Shift+F2



F2 / Shift+F2

Gutter and Lens

Right gutter displays error information. Use Lens mode for preview.



Settings

Enable/disable inspections, see examples and documentation.

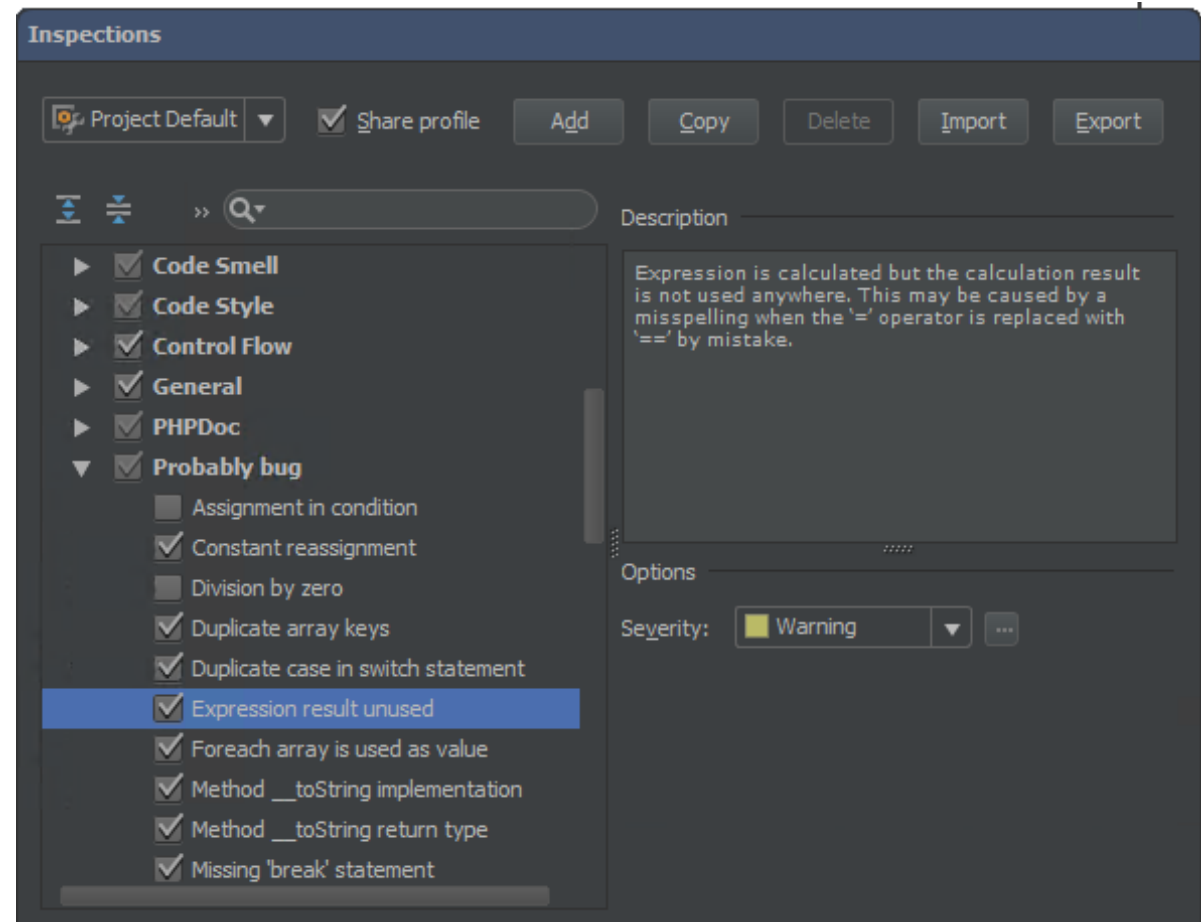
Configure severity.



Ctrl+Alt+S

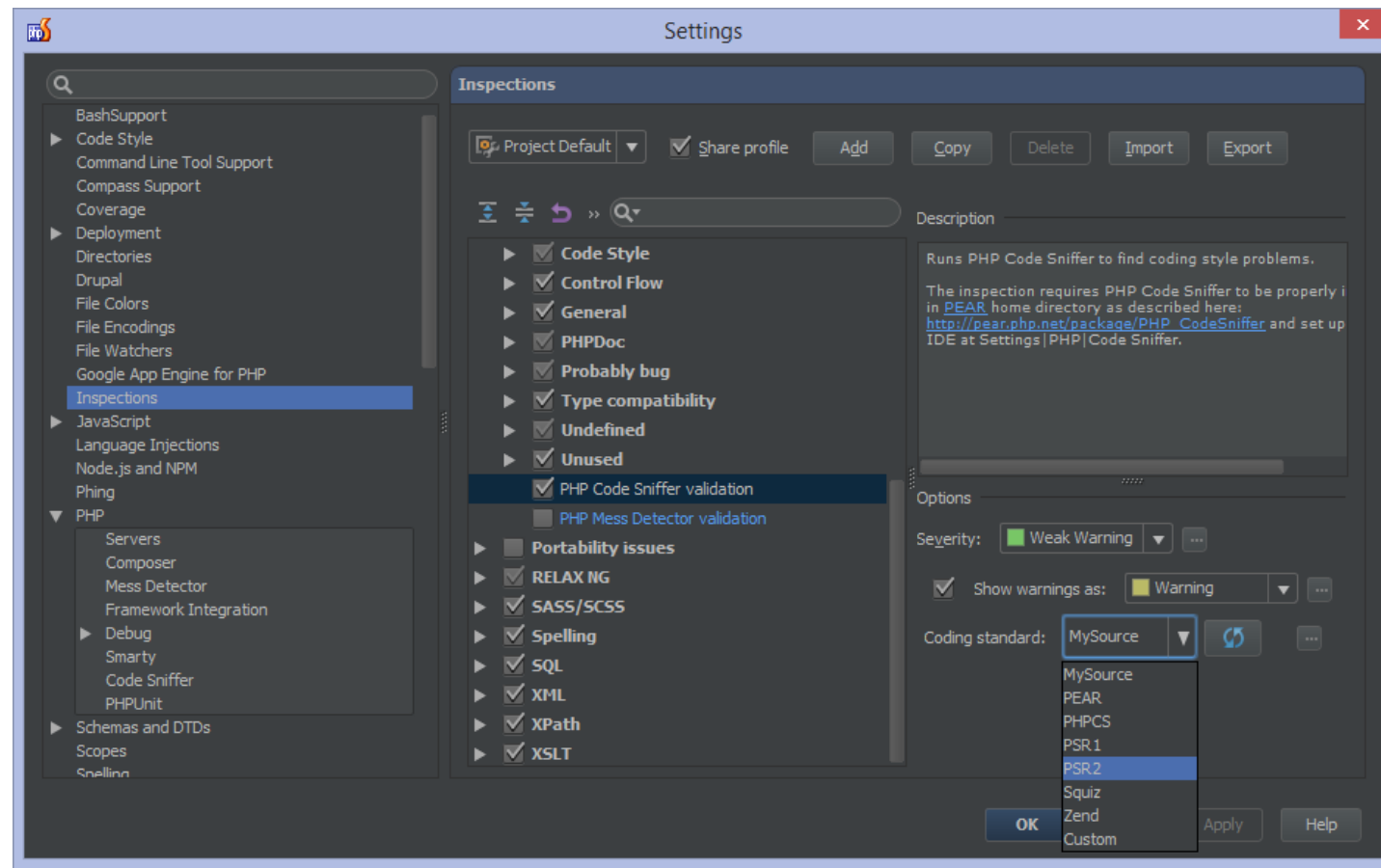


Command+,



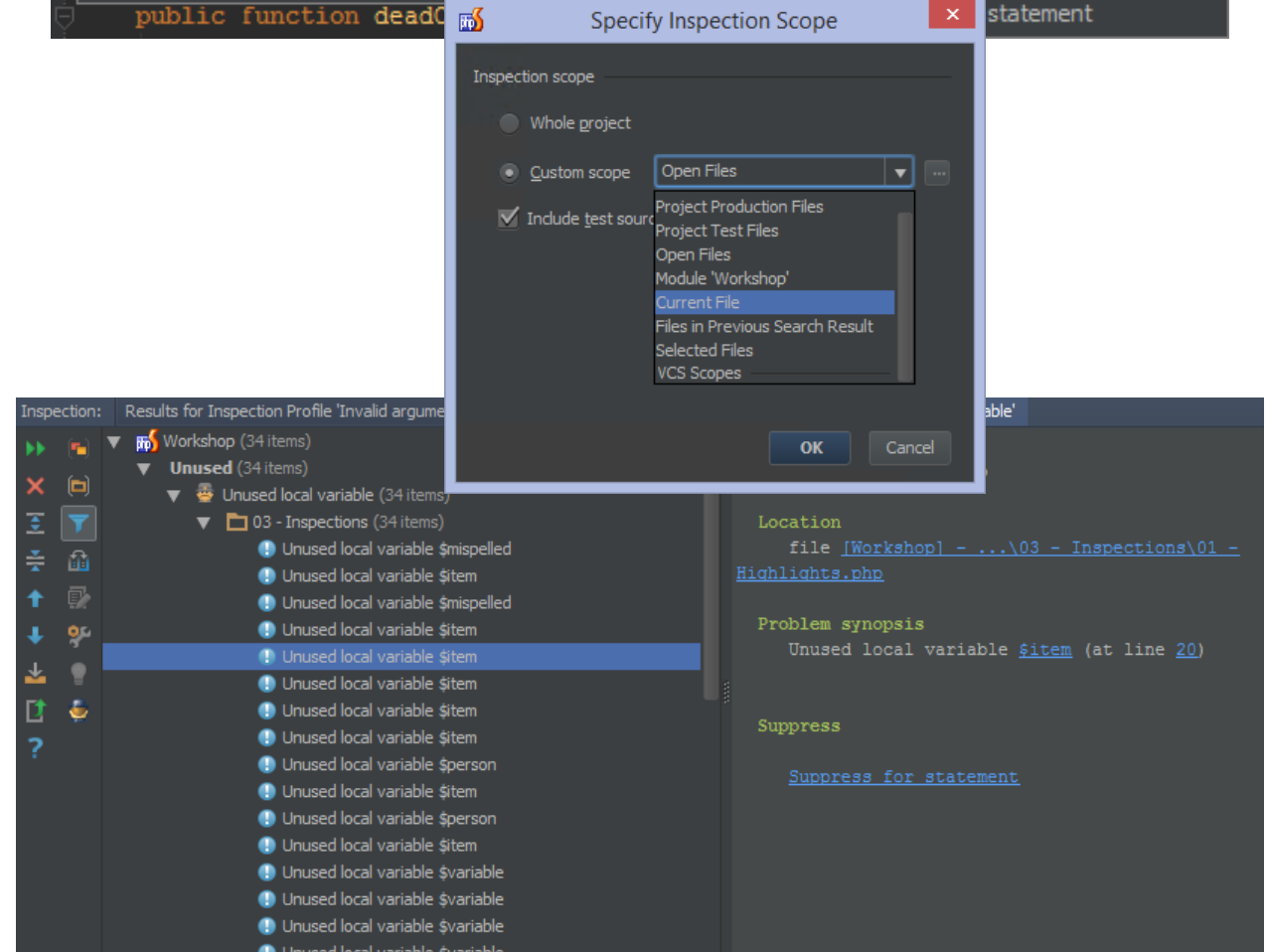
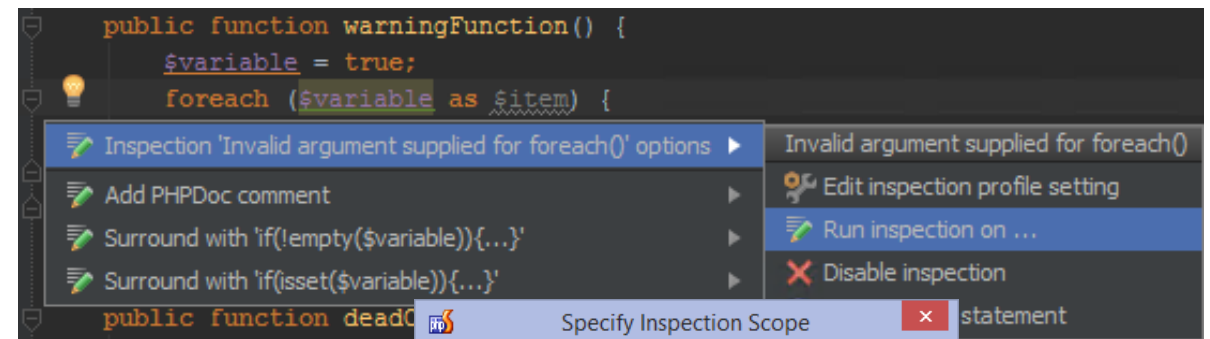
PHPMD / PHPCS

PHP Mess Detector and PHP Code Sniffer can be plugged in for inspections as well.



Find Issues

Quick fix menu allows finding all similar issues.



Run Inspections

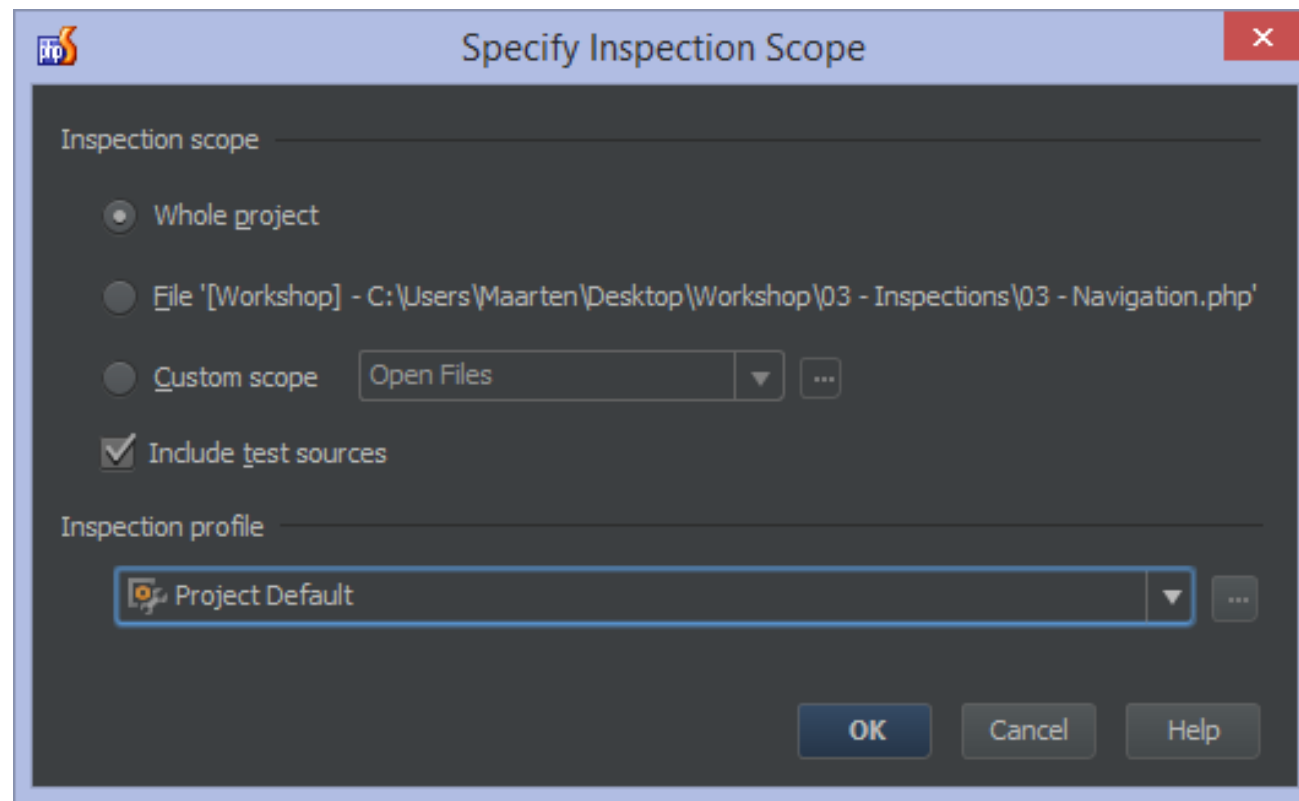
Run an inspection profile and get results for the entire project (or scope).



Ctrl+Alt+Up / Down



Alt+Command+Up / Down



Live templates (snippets)

Code Expansion

Expand text shortcut using template.

Expands into code with variable “hotspots”

Hotspot can be linked to an expression, such as “current user”, “autocomplete”, ...

Tab to move between hotspots.

Template can define end point for caret.

```
$numbers = array(1,2,3,4,5,6,7,8,9);  
  
foreach ($numbers as $item) {  
  
}
```



Creating Live Templates

Create your own live templates.

Use variables and expressions.

Can be created from settings or straight from within the editor.

\$END\$ variable denotes where caret should be after expansion.

Expressions:

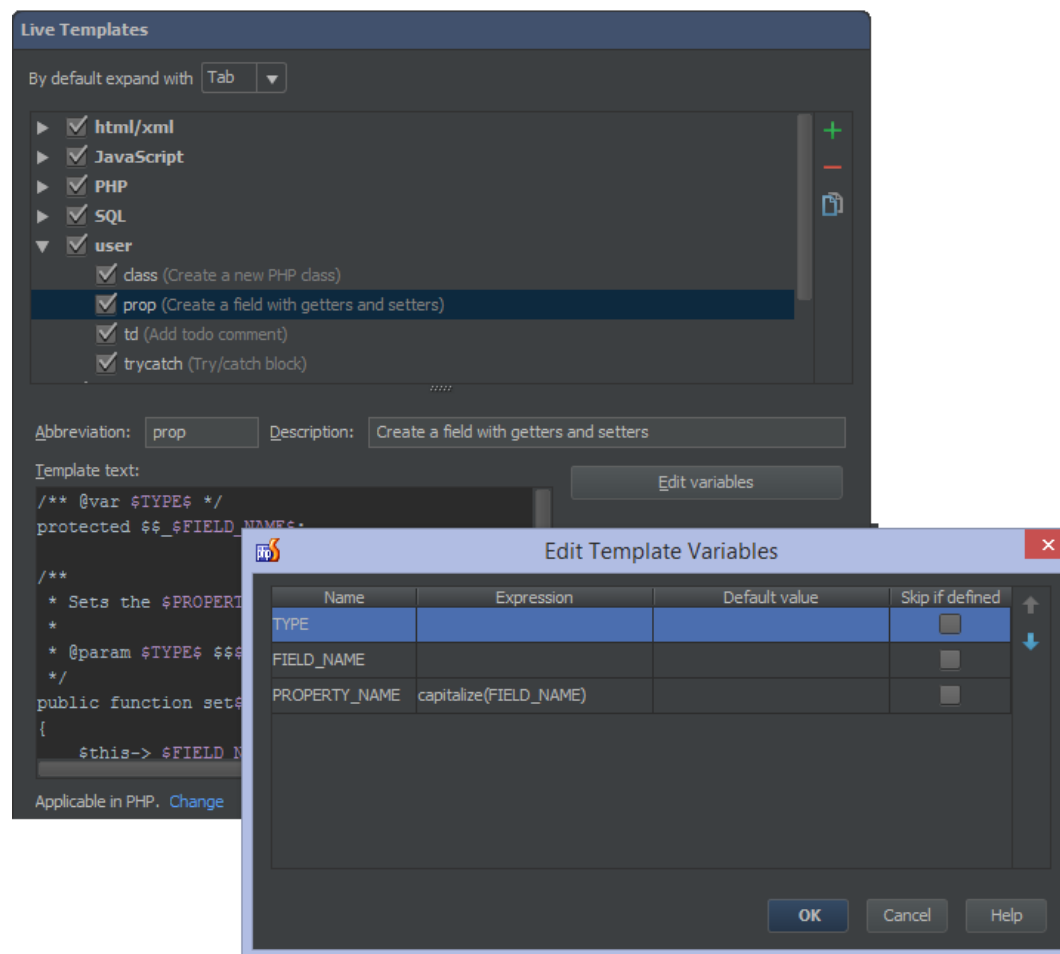
<https://www.jetbrains.com/phpstorm/webhelp/edit-template-variables-dialog.html>



Tab



Tab



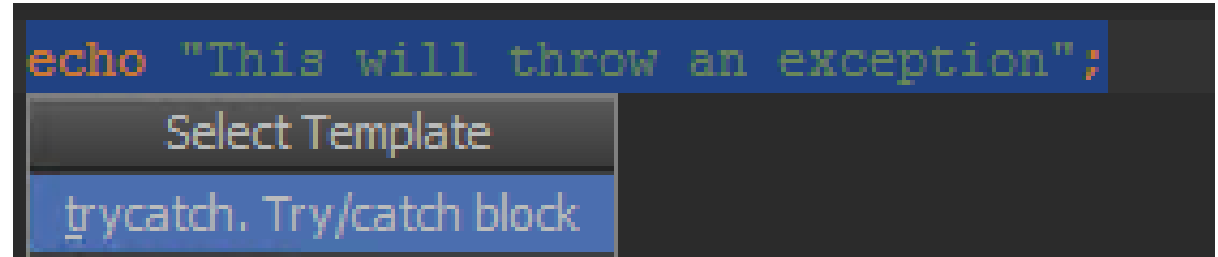
Surround Templates

Surround code with a template.

Extra predefined variable:

`$SELECTION$` is the currently selected text

Only considered a surround template when this macro is used.



Ctrl+Alt+T or Ctrl+Alt+J



Alt+Command+T or Alt+Command+J

File Templates

Applied when creating a new file or generating code.

Can contain includes.

Customize code generation.

Variables:

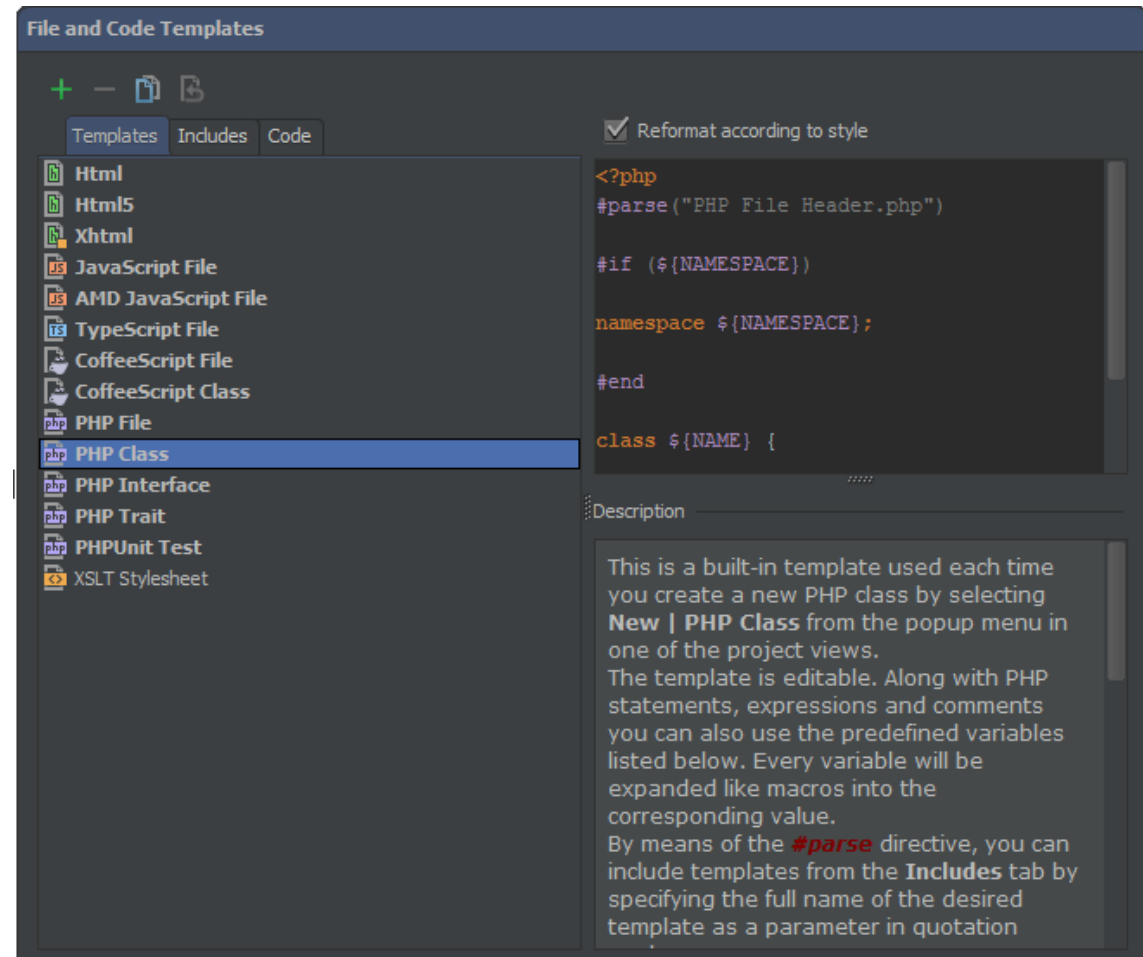
<https://www.jetbrains.com/phpstorm/webhelp/file-template-variables.html>



Ctrl+Alt+S



Command+,



Refactoring

“Refactoring is a controlled technique for improving the design of an existing code base.”

Martin Fowler

<http://martinfowler.com/books/refactoring.html>

Refactoring in PhpStorm

The IDE will:

- Perform the refactoring
- Track down and correct the affected code references automatically
- Warn about occurrences it can not update automatically

Available refactorings:

- Change Signature
- Copy/Clone
- Extract Constant
- Extract Field
- Extract Interface
- Extract Method
- Extract Parameter
- Extract Variable
- Inline
- Move Refactorings
- Pull Members up
- Push Members down
- Rename Refactorings
- Safe Delete
- Move Static Member

Refactor This

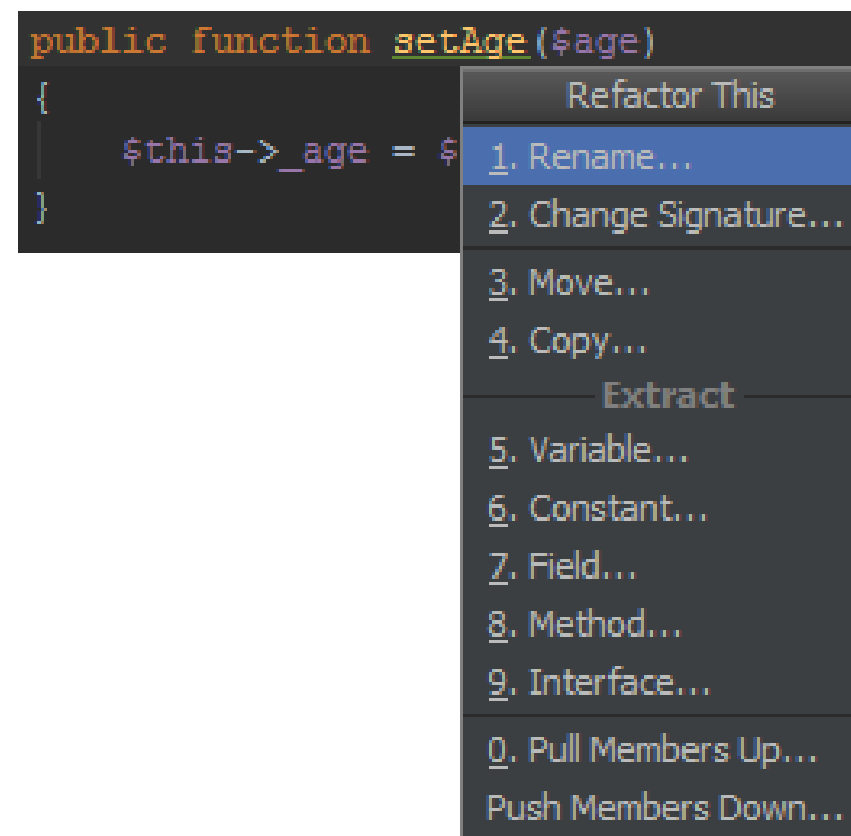
Refactor symbol or code fragment

In Project View, Structure Tool Window, Editor or UML Class Diagram.

Use menu item **Refactor | Refactor This** or keyboard shortcut.

  Ctrl+Shift+Alt+T

 Ctrl+T



Change Signature

<http://www.jetbrains.com/phpstorm/webhelp/change-signature.html>

You can use this refactoring to:

Change the function name.

Add new parameters and remove the existing ones.

Assign default values to the parameters.

Reorder parameters.

Change parameter names.

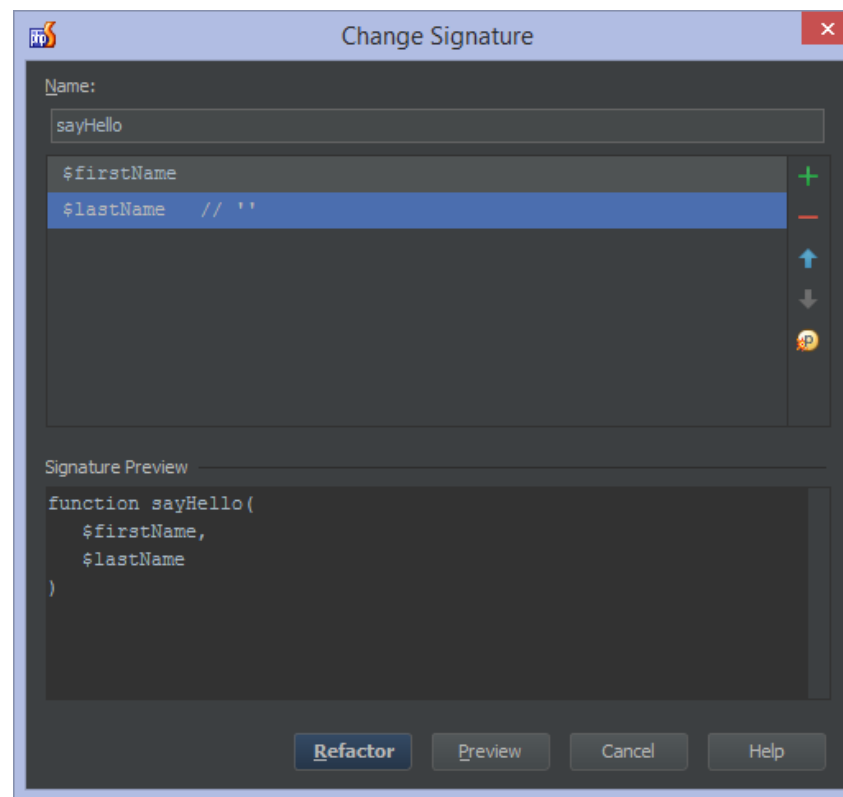
Propagate new parameters through function call hierarchy.



Ctrl+F6



Command+F6



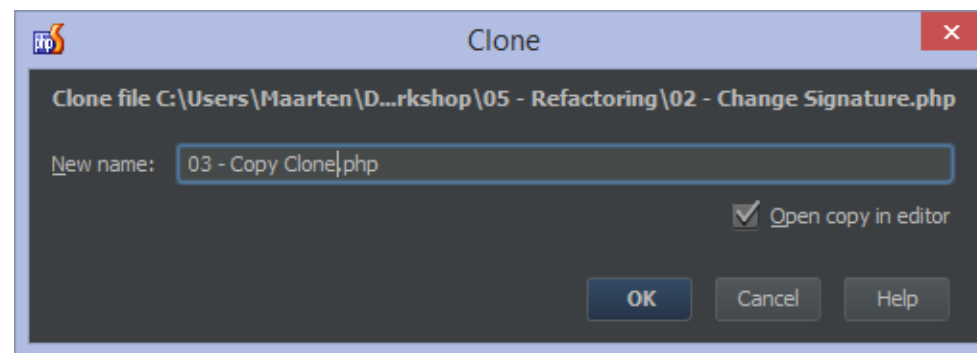
Copy/Clone

<http://www.jetbrains.com/phpstorm/webhelp/copy-clone.html>

Copy a class, file or directory to another directory or clone it within the same directory.

Using keyboard shortcut

Using drag/drop with Ctrl key pressed)



F5 (copy), Shift+F5 (clone)

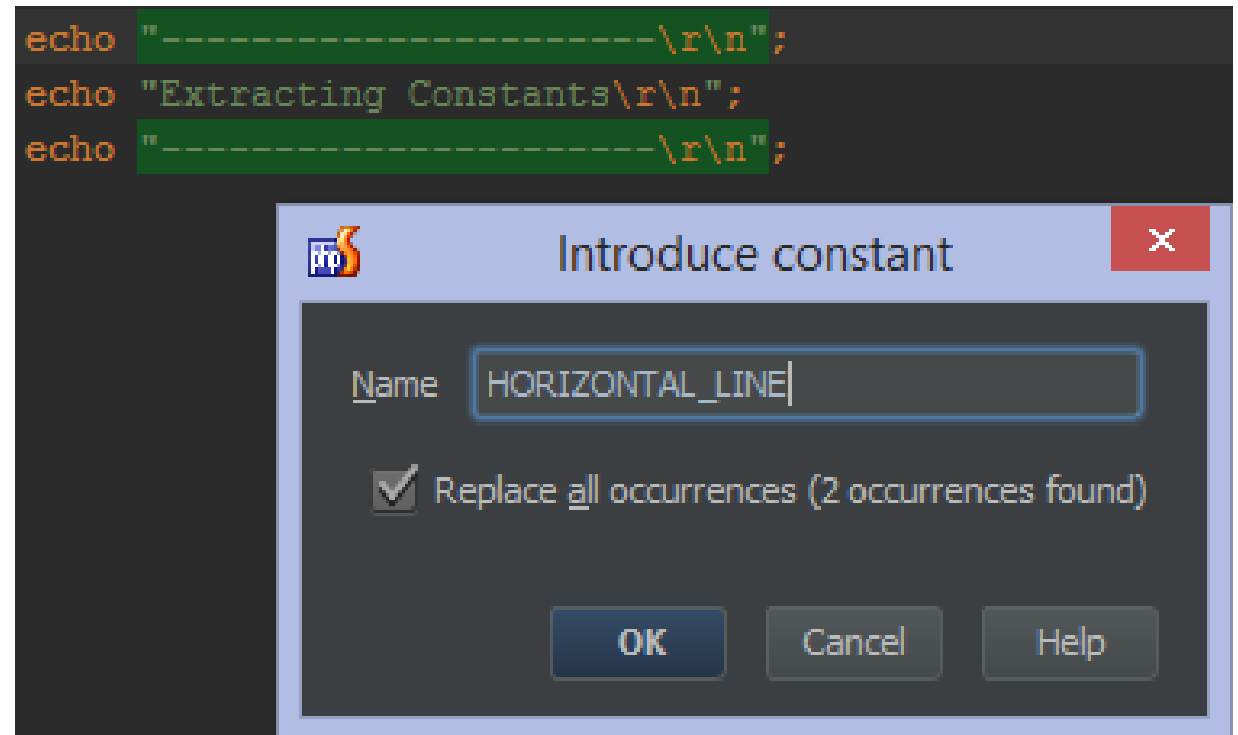
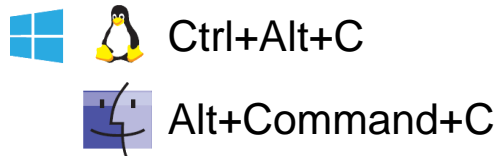


F5 (copy), Shift+F5 (clone)

Extract Constant

<http://www.jetbrains.com/phpstorm/webhelp/extract-constant.html>

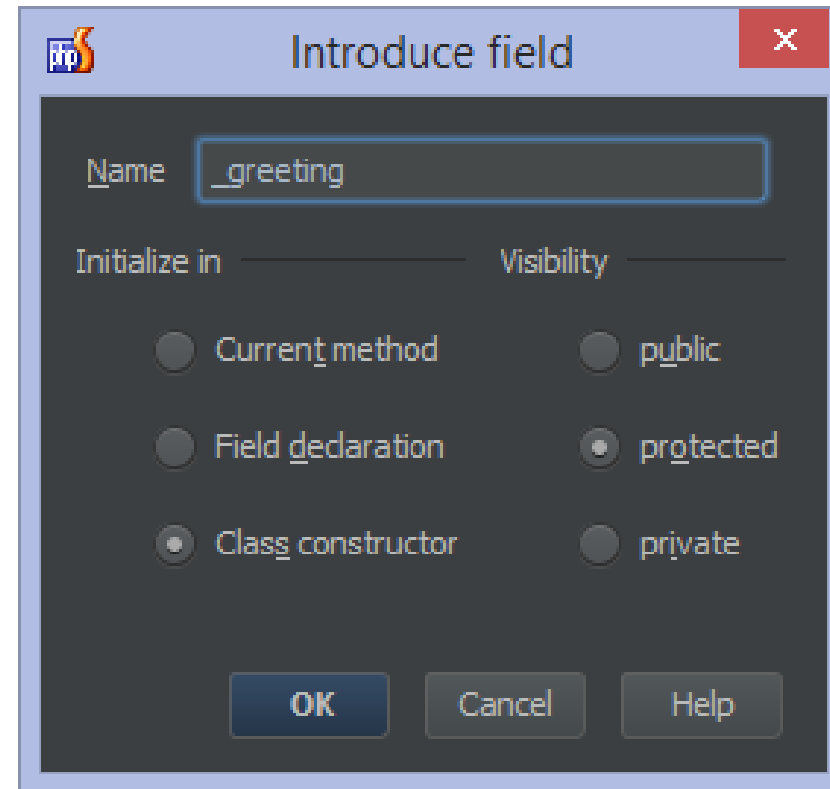
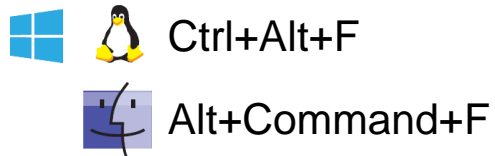
Extract a constant to make code cleaner and more maintainable.



Extract Field

<http://www.jetbrains.com/phpstorm/webhelp/extract-field.html>

Extract an expression into a field.



Extract Interface

<http://www.jetbrains.com/phpstorm/webhelp/extract.html>

Extract an interface from a class.

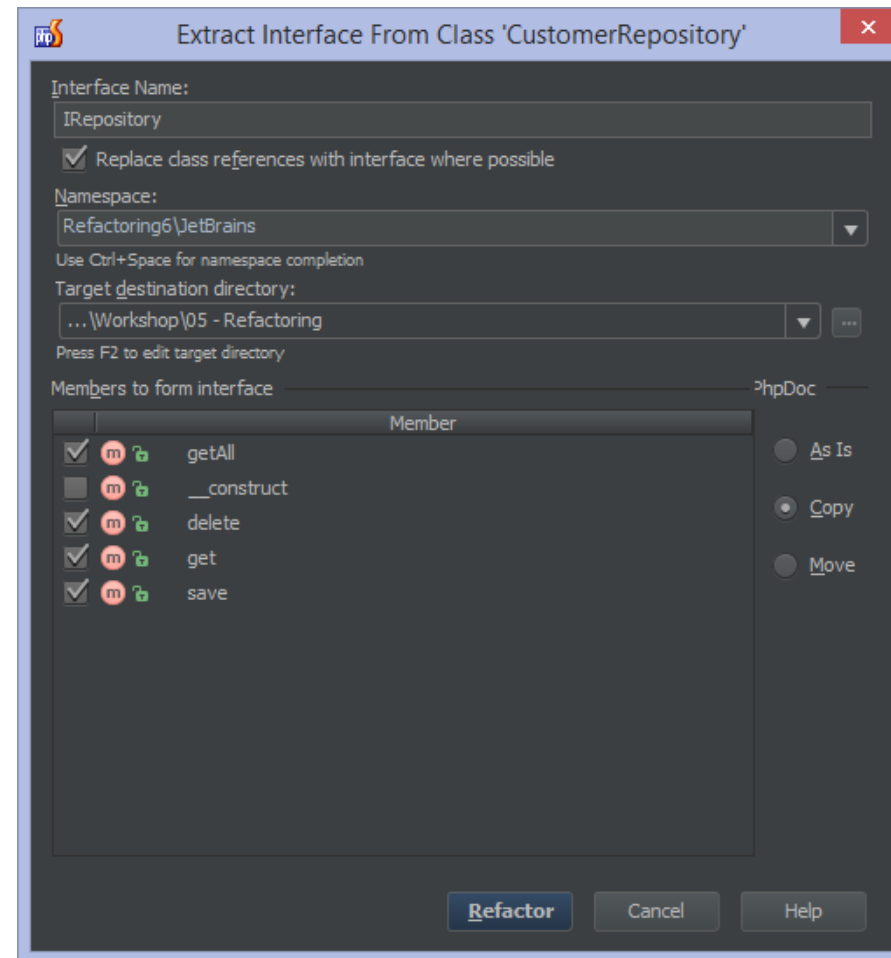
Specify a Name

Optionally change namespace

Pick members to extract

Copy PHPDoc

Use the menu or Refactor This.



Extract Method

<http://www.jetbrains.com/phpstorm/webhelp/extract-method.html>

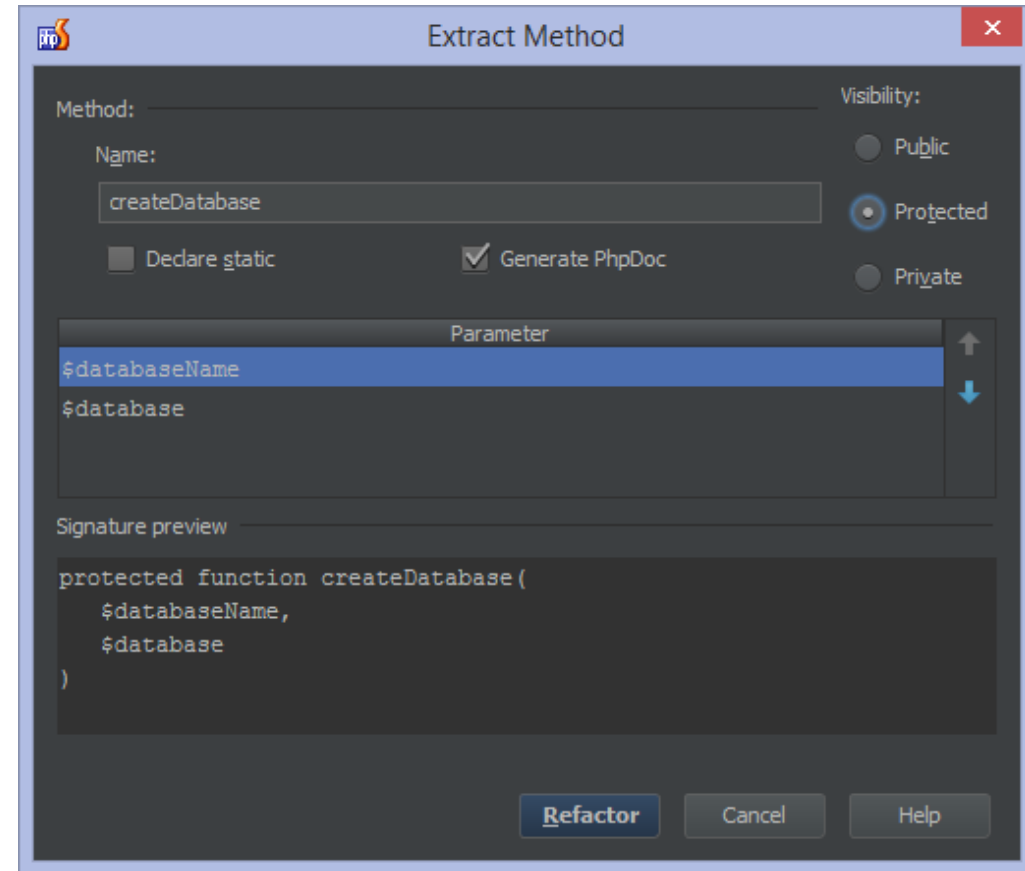
Extracts a block of code into a method, detecting parameters and return values.



Ctrl+Alt+M



Alt+Command+M



Extract Parameter

<http://www.jetbrains.com/phpstorm/webhelp/change-signature.html>

Adds a new parameter to a function declaration. Determines the default. Can generate JSDoc.



Ctrl+Alt+P



Alt+Command+P

```
/**
 * @param (String("Hello")) s
 */
this.greet = function (greeting) {
    // ...
    return greeting + ', ' + _name;
}
```

Generate JSDoc
Optional parameter

Extract Variable

<http://www.jetbrains.com/phpstorm/webhelp/extract-variable.html>

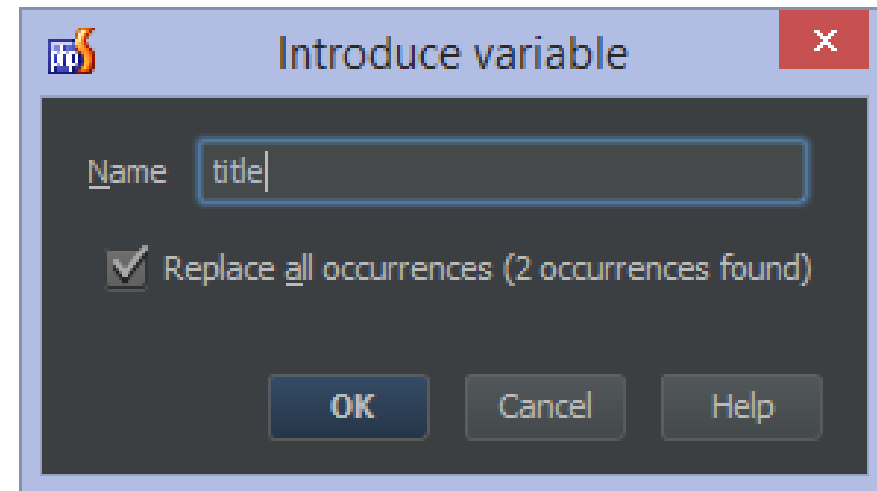
Puts the result of a selected expression into a variable. The original expression is replaced with the new variable.



Ctrl+Alt+V



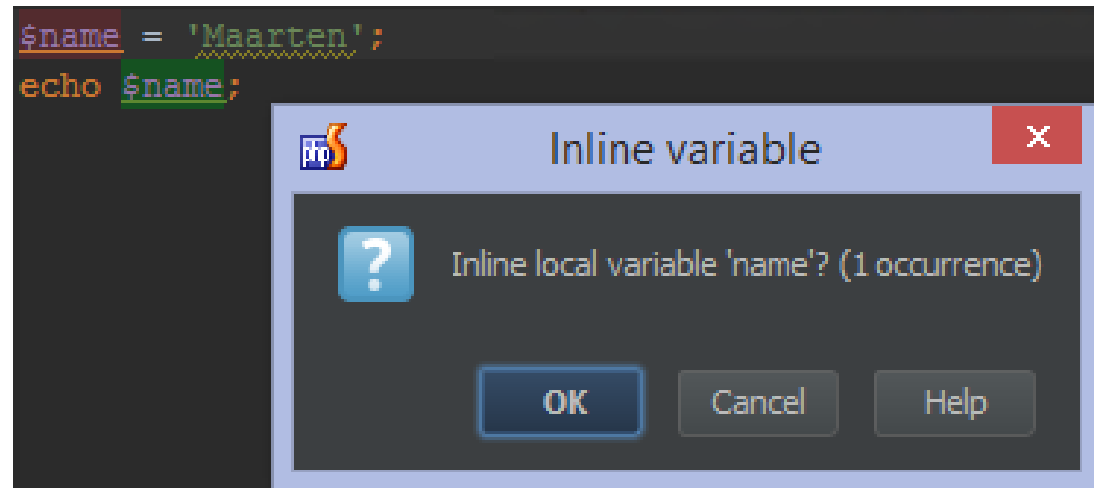
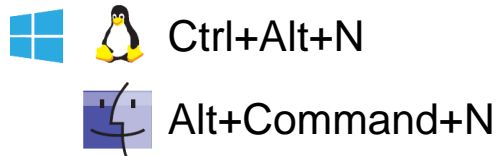
Alt+Command+V



Inline

<http://www.jetbrains.com/phpstorm/webhelp/inline.html>

Replace redundant variables or functions with the full expression. It is the opposite of **Extract Variable / Method**.



Move

<http://www.jetbrains.com/phpstorm/webhelp/move-refactorings.html>

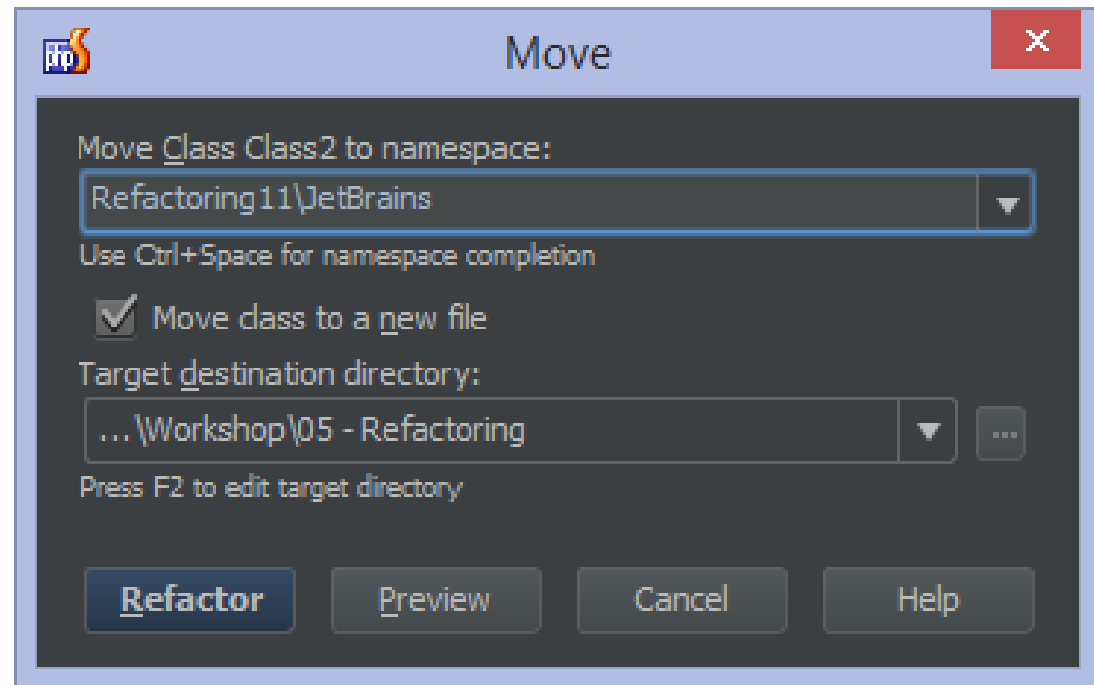
Changes the location of a file, directory, class or static member.

Move File

Move Directory

Move Class

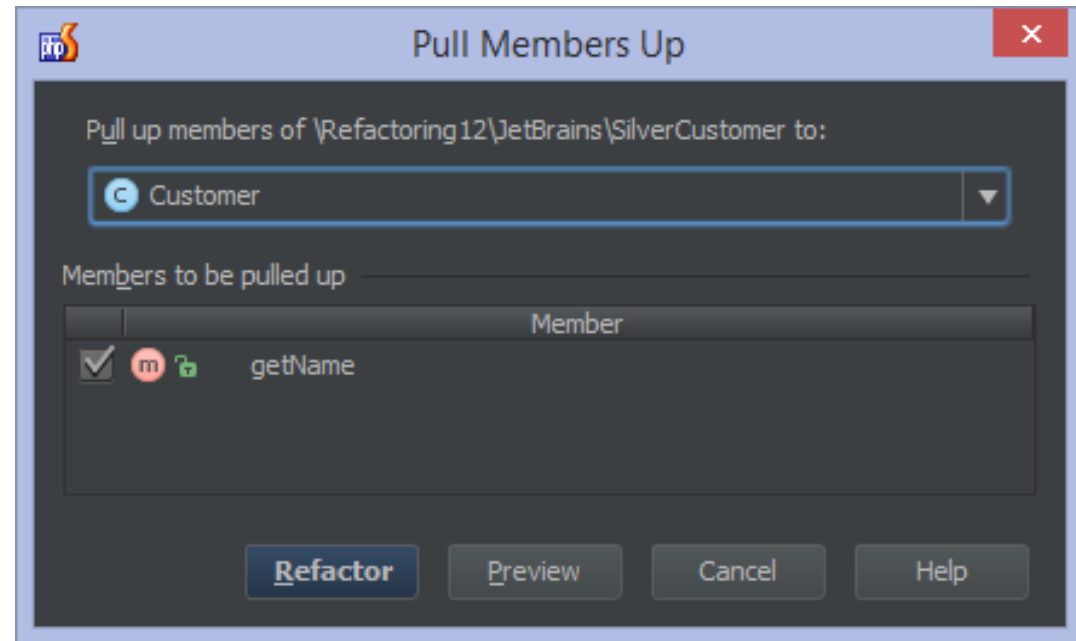
Move Static Member



Pull Members Up / Push Members Down

<http://www.jetbrains.com/phpstorm/webhelp/pull-members-up.html>

Move members from subclass to superclass or from superclass to subclass.



Rename

<http://www.jetbrains.com/phpstorm/webhelp/rename-refactorings.html>

Allow you to rename symbols,
automatically correcting all references
in the code.

Rename Class

Rename Method

Rename Field

Rename Function

Rename Variable

Rename Parameter

Rename CSS color value

Rename File

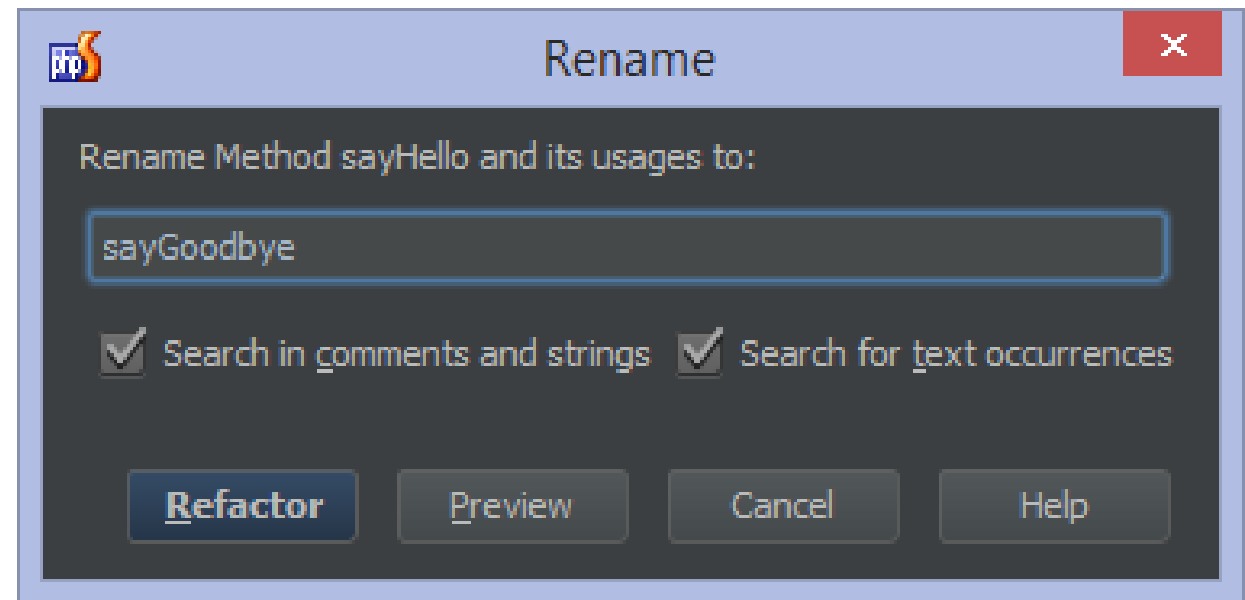
Rename Directory



Shift+F6



Shift+F6




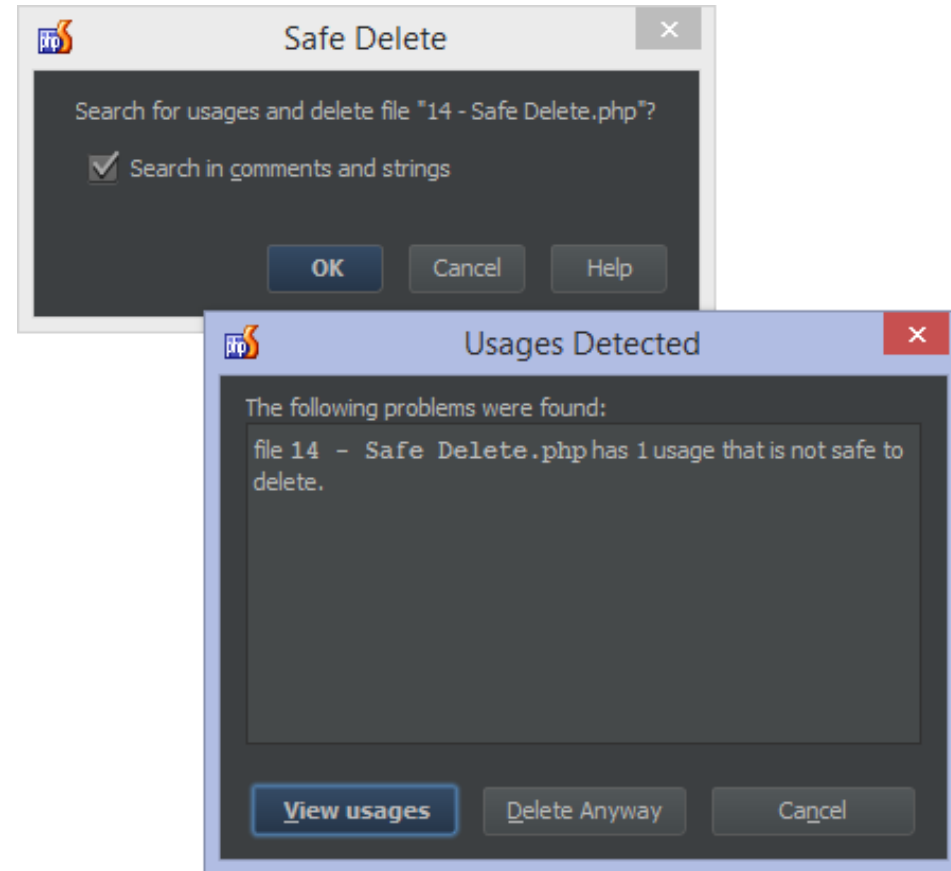
Safe Delete

<http://www.jetbrains.com/phpstorm/webhelp/safe-delete.html>

Safely remove code or symbols.

  Alt+Delete

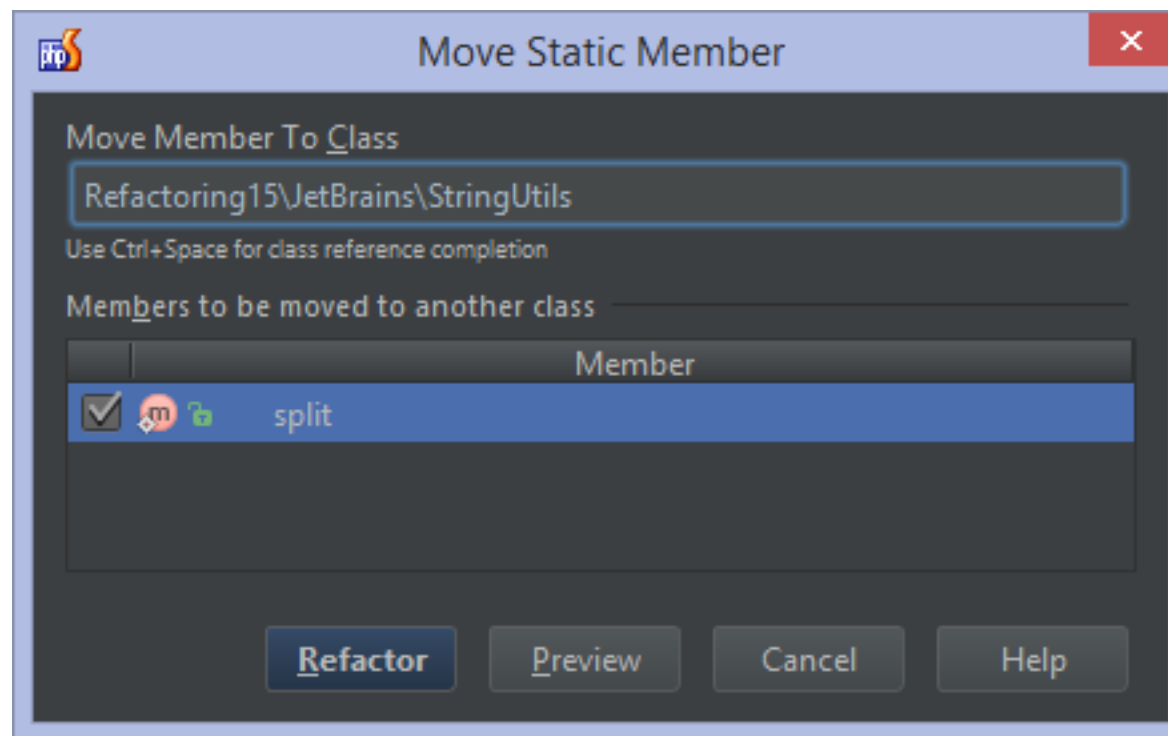
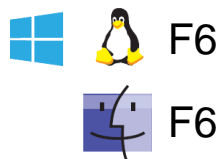
 Command+Delete



Move Static Member

<http://www.jetbrains.com/phpstorm/webhelp/move-refactorings.html#d705080e490>

Move static fields and methods to another type.



Debugging

Debugging

“Finding and reducing bugs”

In an IDE typically stepping through code & inspecting values in memory

PhpStorm needs one of these configured:

Xdebug - <http://confluence.jetbrains.com/display/PhpStorm/Xdebug+Installation+Guide>

Zend Debugger -

<http://confluence.jetbrains.com/display/PhpStorm/Zend+Debugger+Installation+Guide>

Most features supported by both debuggers

Settings

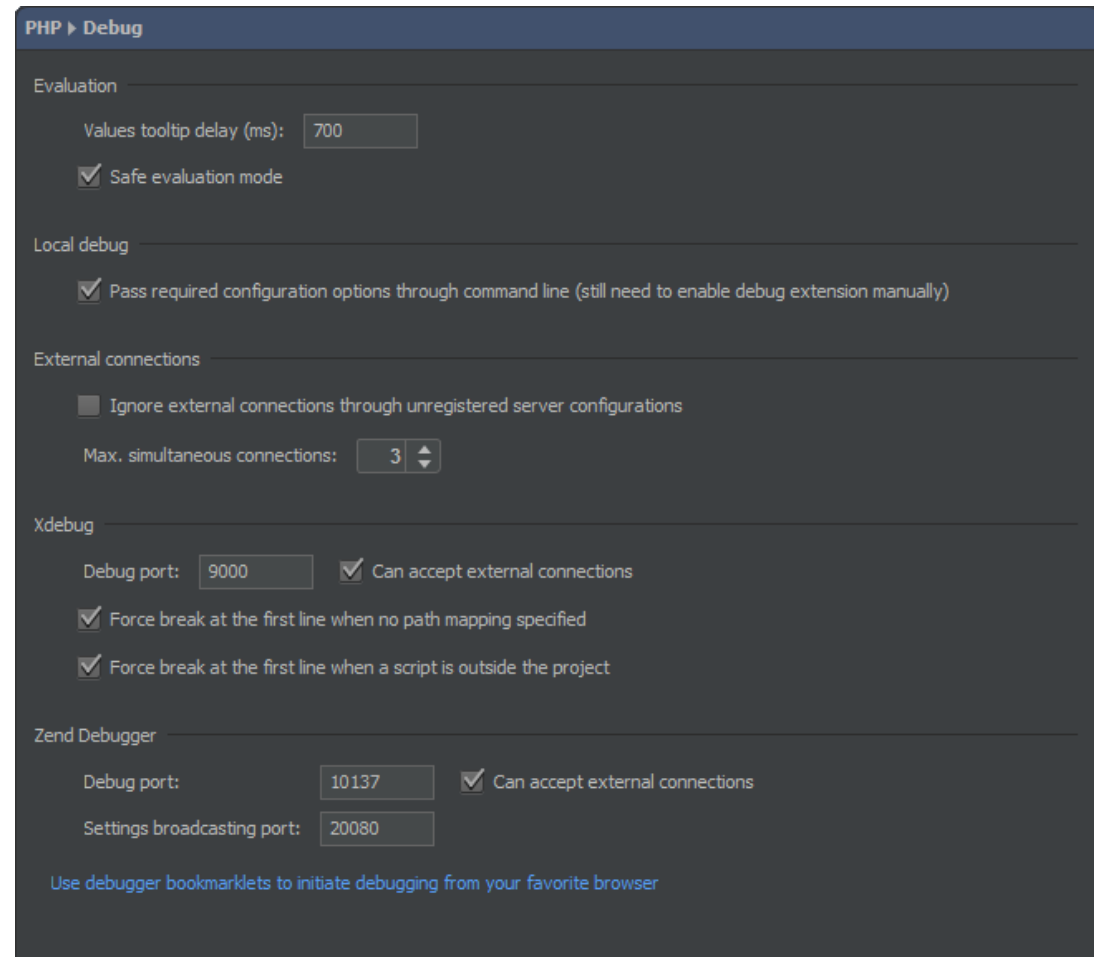
Debugger is configured through
php.ini

Project Settings | PHP | Debug

How debug info is displayed in IDE

Xdebug / Zend Debugger port
number

DBGp Proxy

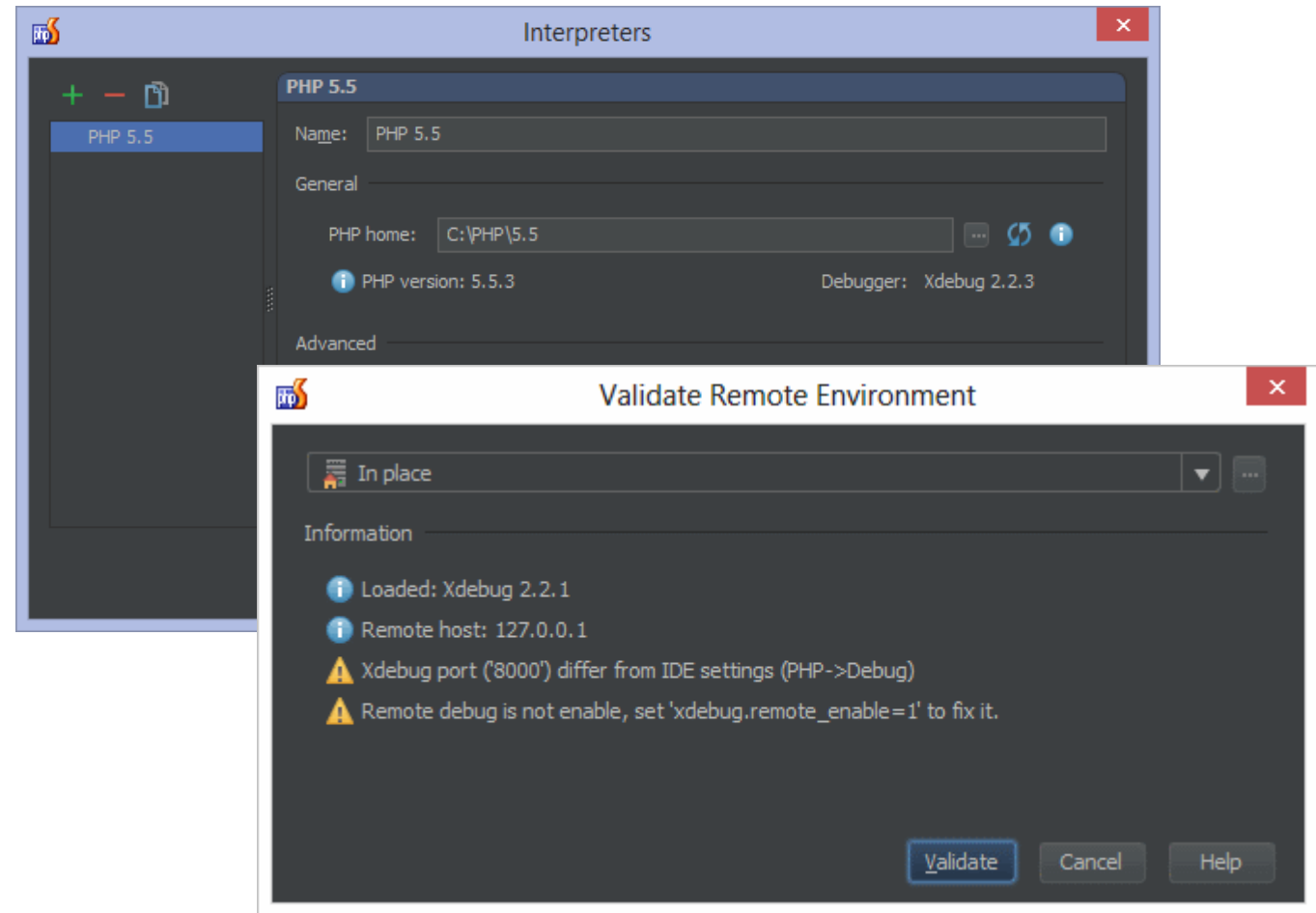


Debugger Configuration Validation

PhpStorm analyzes
configuration

Reports active debugger

When a server is configured, reports
configuration issues



Breakpoints

Tell the interpreter to pause execution and inspect variables.

```
● $name = 'Mikhail';  
- for ($i = 0; $i < 5; $i++) {  
● $name = 'Person ' . $i;  
- }
```



Ctrl+F8



Command+F8

Conditional Breakpoints

Tell the interpreter to pause execution and inspect variables, only when a specific condition is true.

```
● $name = 'Mikhail';  
- for ($i = 0; $i < 5; $i++) {  
● $name = 'Person ' . $i;  
- }
```



Ctrl+Shift+F8



Shift+Command+F8

Breakpoints

See breakpoints that are specified and configure additional options.

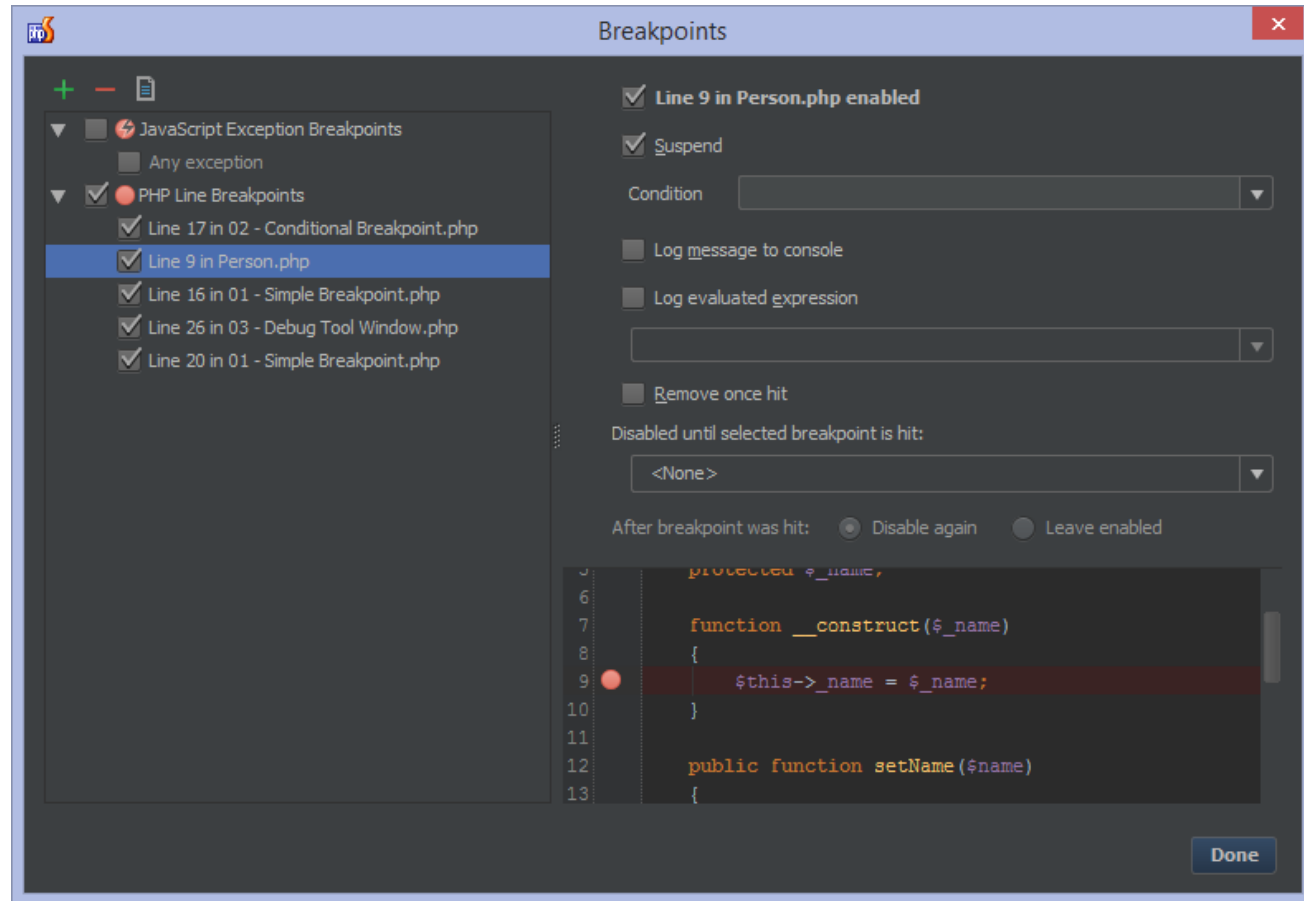
Log a message to console

Remove the breakpoint after it has been hit

Disable breakpoint until another breakpoint has been hit

  Ctrl+Shift+F8

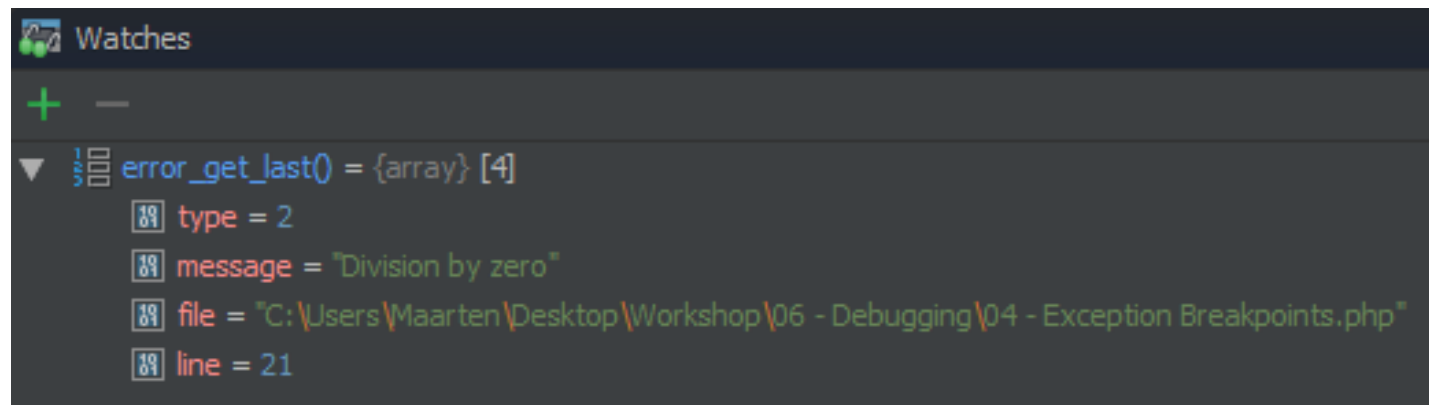
 Shift+Command+F8



Exception Breakpoints

Xdebug only!

Break when error, warning, notice or Exception occurs.



```
Watches
+ -
▼ error_get_last() = {array} [4]
    type = 2
    message = "Division by zero"
    file = "C:\Users\Maarten\Desktop\Workshop\06 - Debugging\04 - Exception Breakpoints.php"
    line = 21
```



Ctrl+Shift+F8



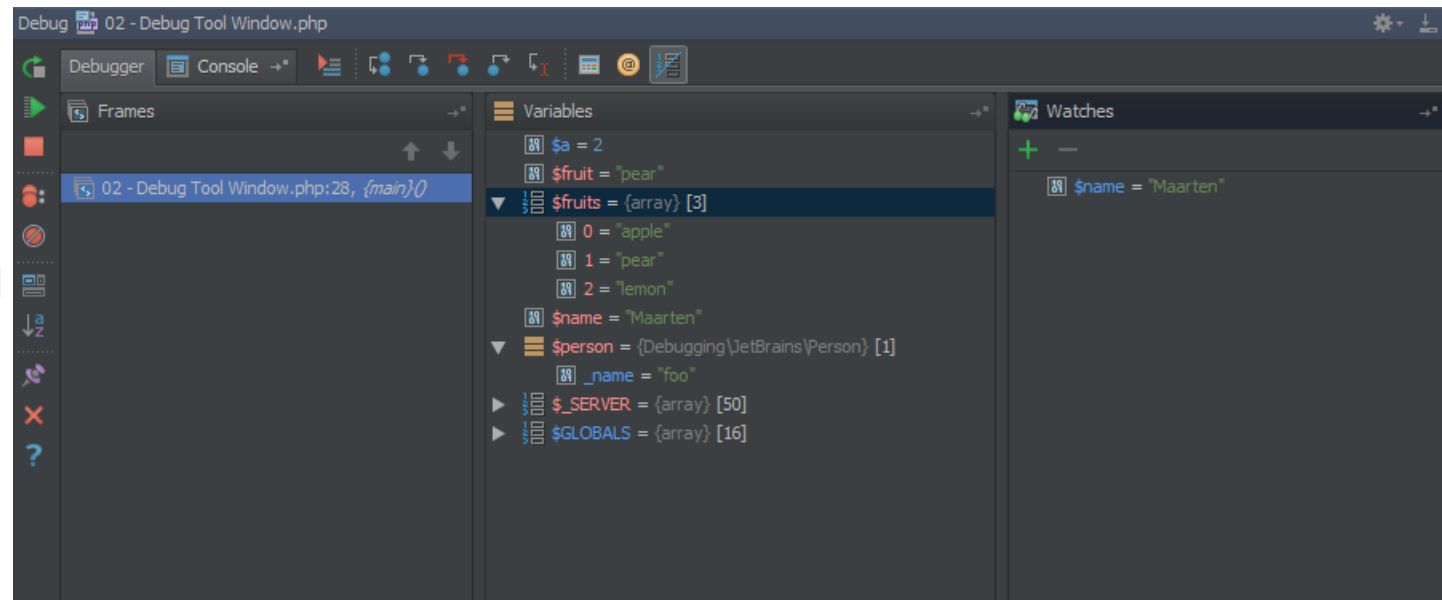
Shift+Command+F8

Debug Tool Window

Available during a debug session.

Showing execution details, variables, watches.

Allows running code / modifying variable value.



Alt+5, Alt+F8 evaluate expression



Command+5, Alt+F8 evaluate expression

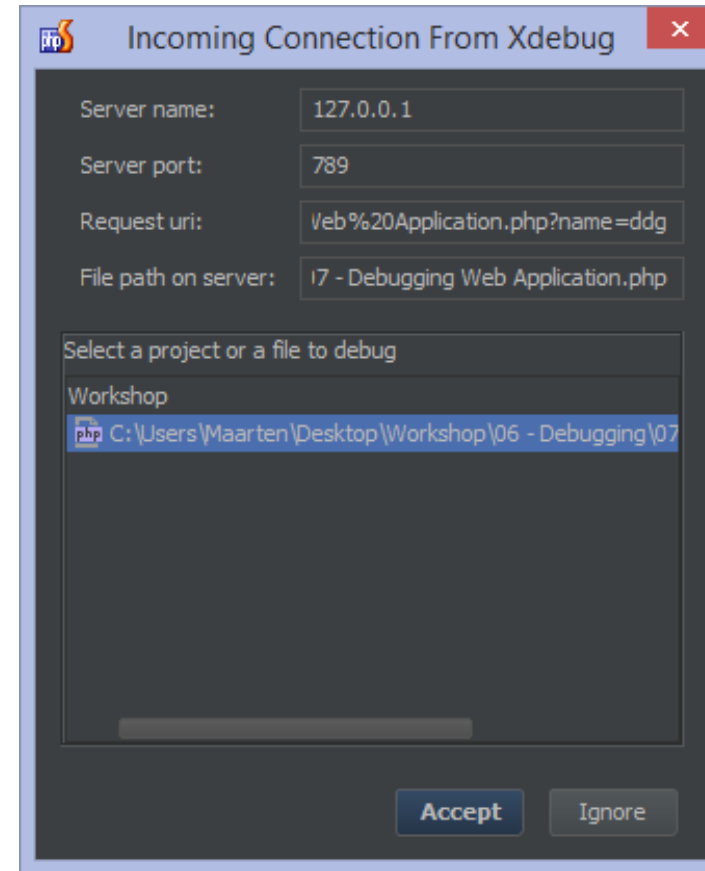
Debugging a Web Application

Use a Run Configuration

PHP HTTP Request – automate the request

PHP Web Application – start with debugging

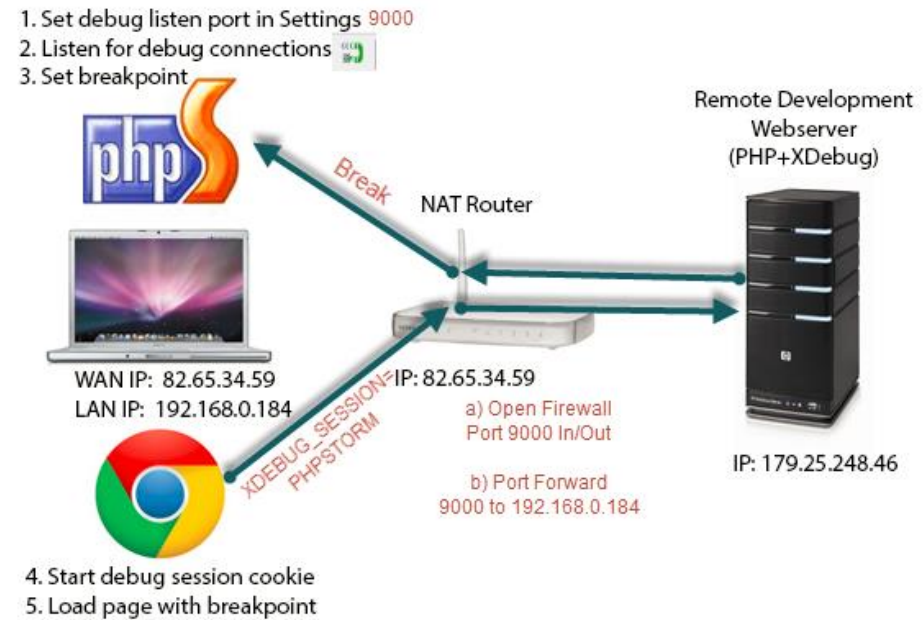
Listen for incoming connections and use bookmarklets or plugin



DBGp Proxy

Debug on remote server.

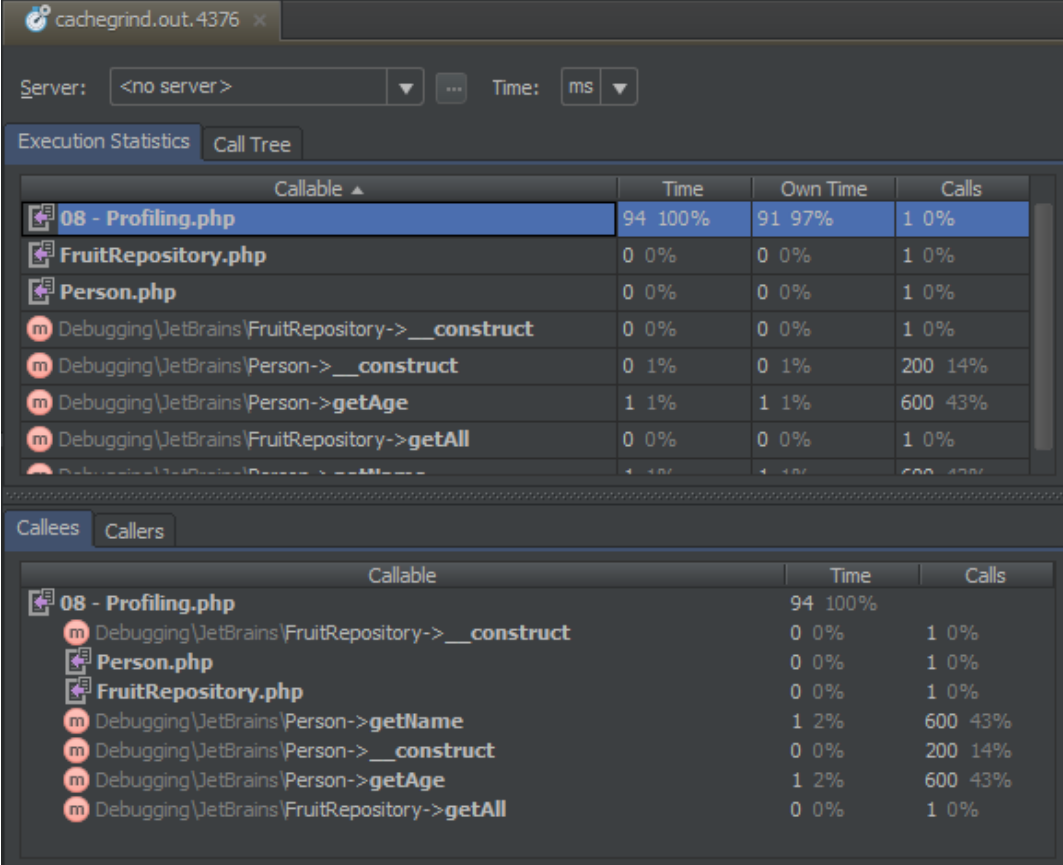
Every developer can use different IDE key and debug independent from each other.



From <http://matthardy.net/blog/configuring-phpstorm-xdebug-dbgp-proxy-settings-remote-debugging-multiple-users/>

Profiling

Insight into # of calls, execution times per function, ...



The screenshot shows the PhpStorm Profiling tool interface. At the top, there's a tab for 'cachegrind.out.4376'. Below it, a 'Server' dropdown is set to '<no server>' and a 'Time' unit is set to 'ms'. Two tabs are visible: 'Execution Statistics' (selected) and 'Call Tree'. The 'Execution Statistics' tab displays a table with columns: Callable, Time, Own Time, and Calls. The 'Callees' tab is also visible below it, showing a similar table for the callees of the selected function.

Callable	Time	Own Time	Calls
08 - Profiling.php	94 100%	91 97%	1 0%
FruitRepository.php	0 0%	0 0%	1 0%
Person.php	0 0%	0 0%	1 0%
Debugging\JetBrains\FruitRepository->__construct	0 0%	0 0%	1 0%
Debugging\JetBrains\Person->__construct	0 1%	0 1%	200 14%
Debugging\JetBrains\Person->getAge	1 1%	1 1%	600 43%
Debugging\JetBrains\FruitRepository->getAll	0 0%	0 0%	1 0%

Callable	Time	Calls
08 - Profiling.php	94 100%	
Debugging\JetBrains\FruitRepository->__construct	0 0%	1 0%
Person.php	0 0%	1 0%
FruitRepository.php	0 0%	1 0%
Debugging\JetBrains\Person->getName	1 2%	600 43%
Debugging\JetBrains\Person->__construct	0 0%	200 14%
Debugging\JetBrains\Person->getAge	1 2%	600 43%
Debugging\JetBrains\FruitRepository->getAll	0 0%	1 0%

Todo explorer

Highlighting TODO

Listing tasks commented in code.

Work in every file type.

Default patterns:

`// todo`

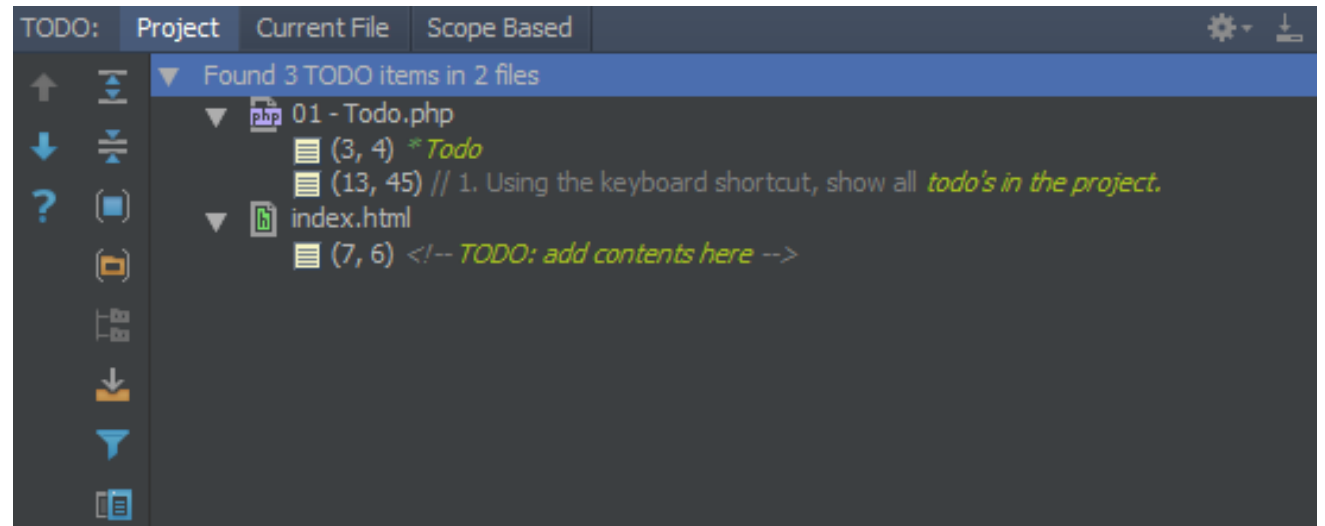
`// fixme`



Alt+6

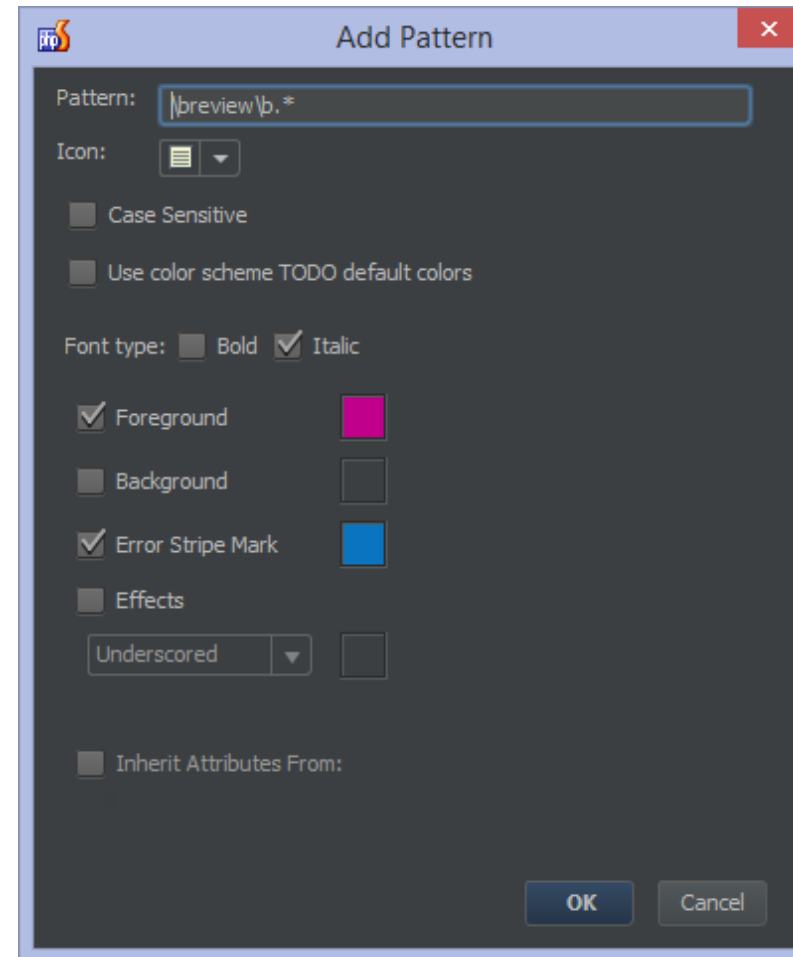


Command+6



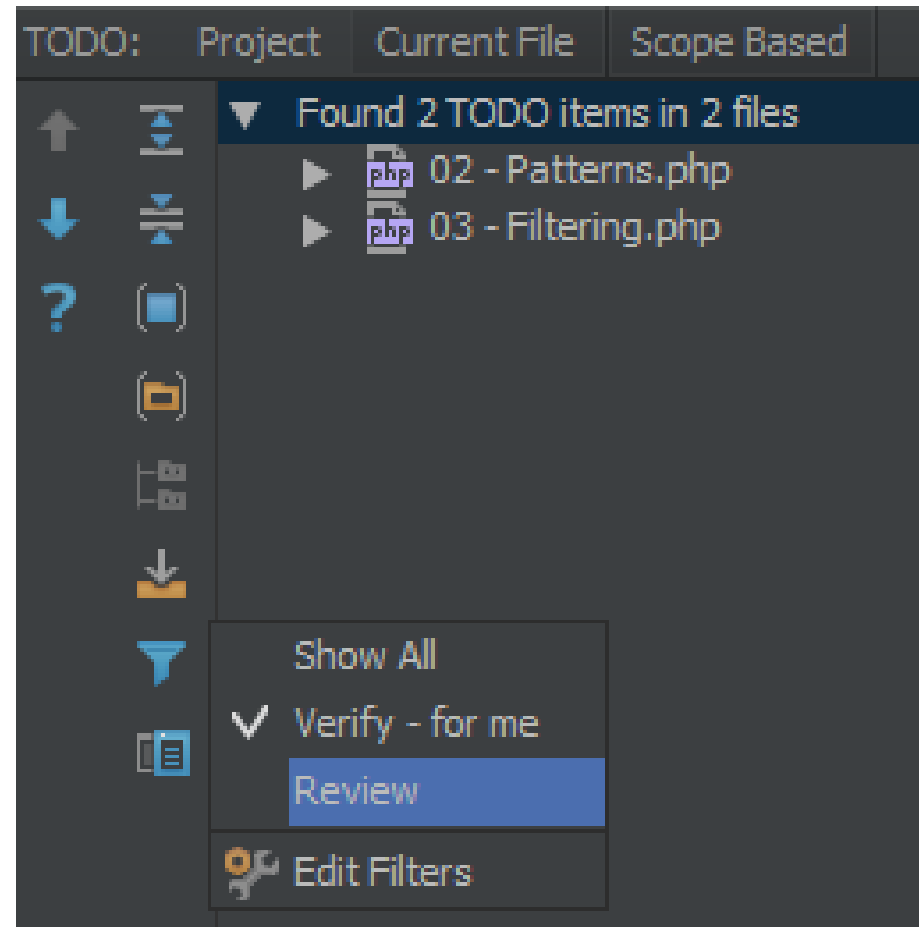
Patterns

Customize todo patterns to recognize other keywords.



Filtering

Filter tasks in the tool window based on pattern.



Unit Testing

Writing PHPUnit tests

Create new file, select PHPUnit test.

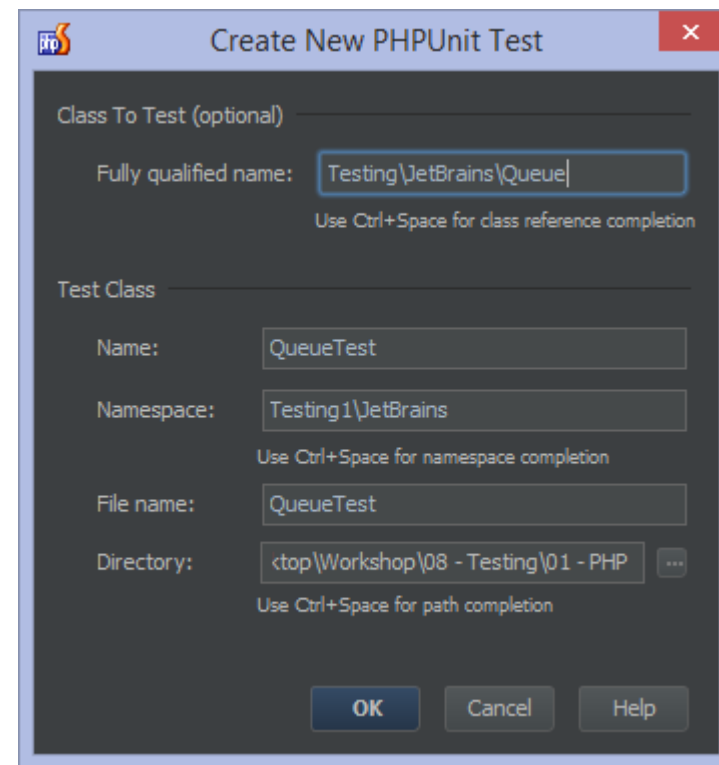
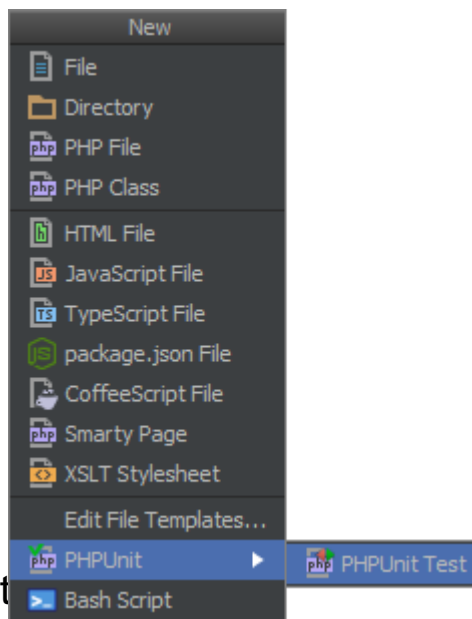
Specify unit test details.



Alt+Insert or Ctrl+Shift+T (Go to Test)



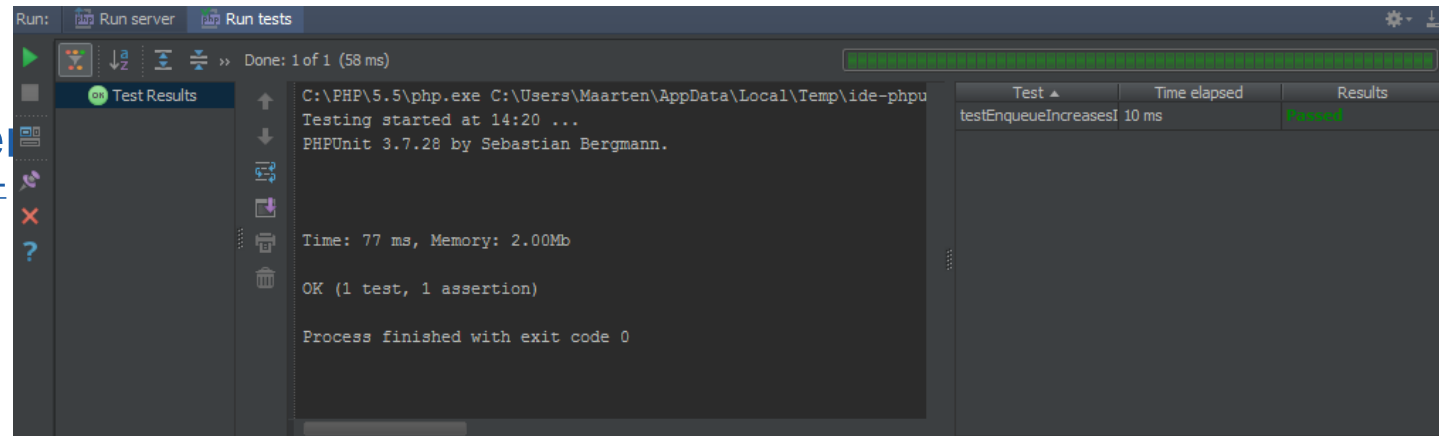
Command+N or Command+Shift+T (Go to Test)



Running PHPUnit tests

Run Unit Tests using PHPUnit.
Note that the PHPUnit framework can be acquired through PEAR or Composer.

Can be run local or on remote server
(see <http://www.jetbrains.com/phpstorm/webhelp/run-debug-configuration-phpunit-on-server.html>)



The screenshot shows the 'Run' tab in an IDE, specifically the 'Run tests' sub-tab. The console output displays the following information:

```
C:\PHP\5.5\php.exe C:\Users\Maarten\AppData\Local\Temp\ide-phpu
Testing started at 14:20 ...
PHPUnit 3.7.28 by Sebastian Bergmann.

Time: 77 ms, Memory: 2.00Mb

OK (1 test, 1 assertion)

Process finished with exit code 0
```

On the right side of the console, there is a table summarizing the test results:

Test	Time elapsed	Results
testEnqueueIncreasesI	10 ms	Passed

Test-Driven Development (TDD)

Writing tests first, outlining expected results, after which the method under test gets implemented.

```
public function testPeekReturnsFalseWhenNoItemsInQueue() {  
    $result = $this->_queue->peek();  
    assertFalse($result);  
}
```

  Alt+Enter

 Alt+Enter

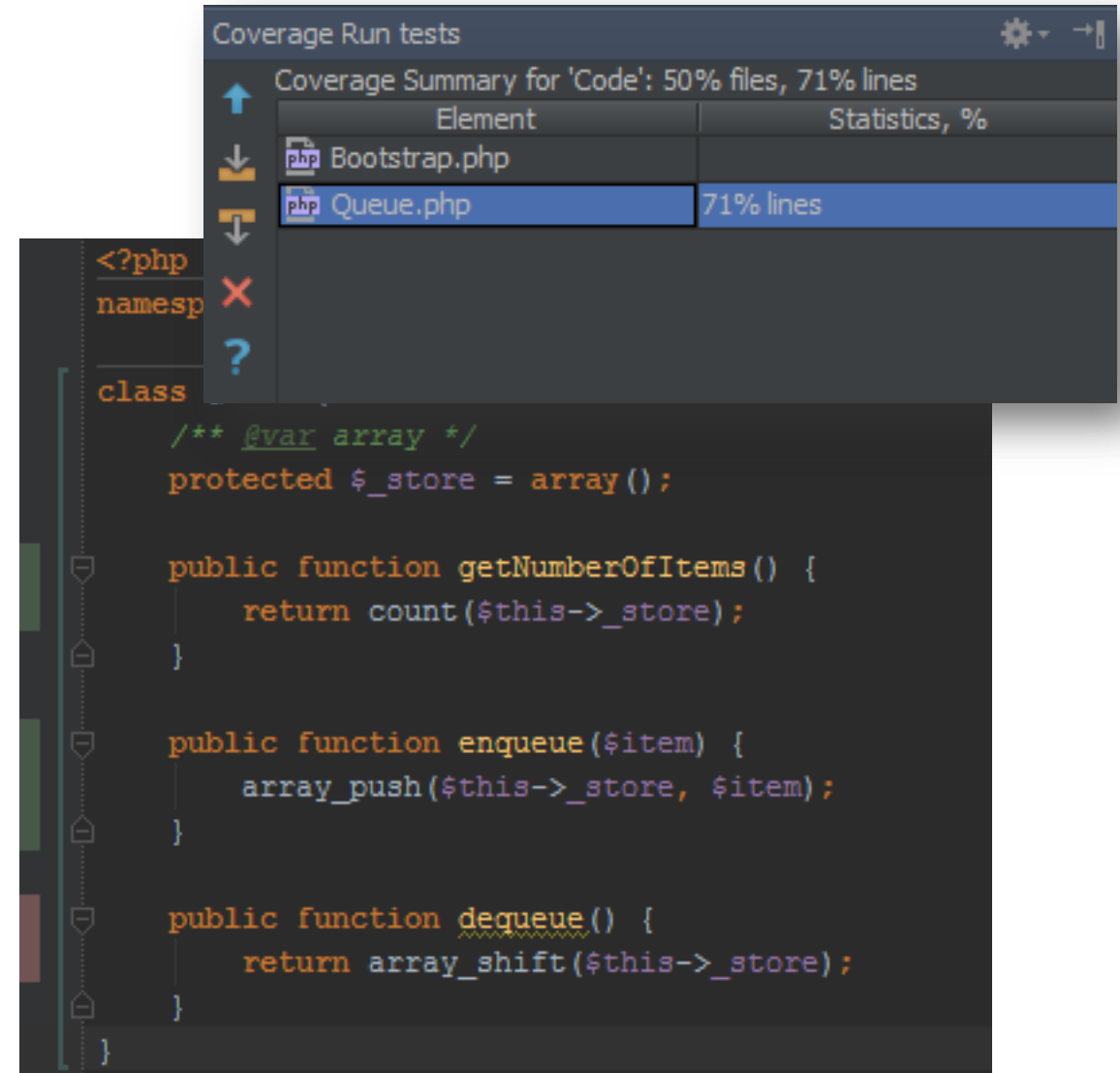
Code Coverage

See which statements have been tested and which statements have not.

Use the “Run tests with Coverage” action.

See

<https://www.jetbrains.com/phpstorm/webhelp/coverage.html>



JavaScript Unit Testing

May require a plugin to be installed, e.g. [Karma](#)

Using Karma or JSTestDriver.

[Installing Karma through NPM](#)

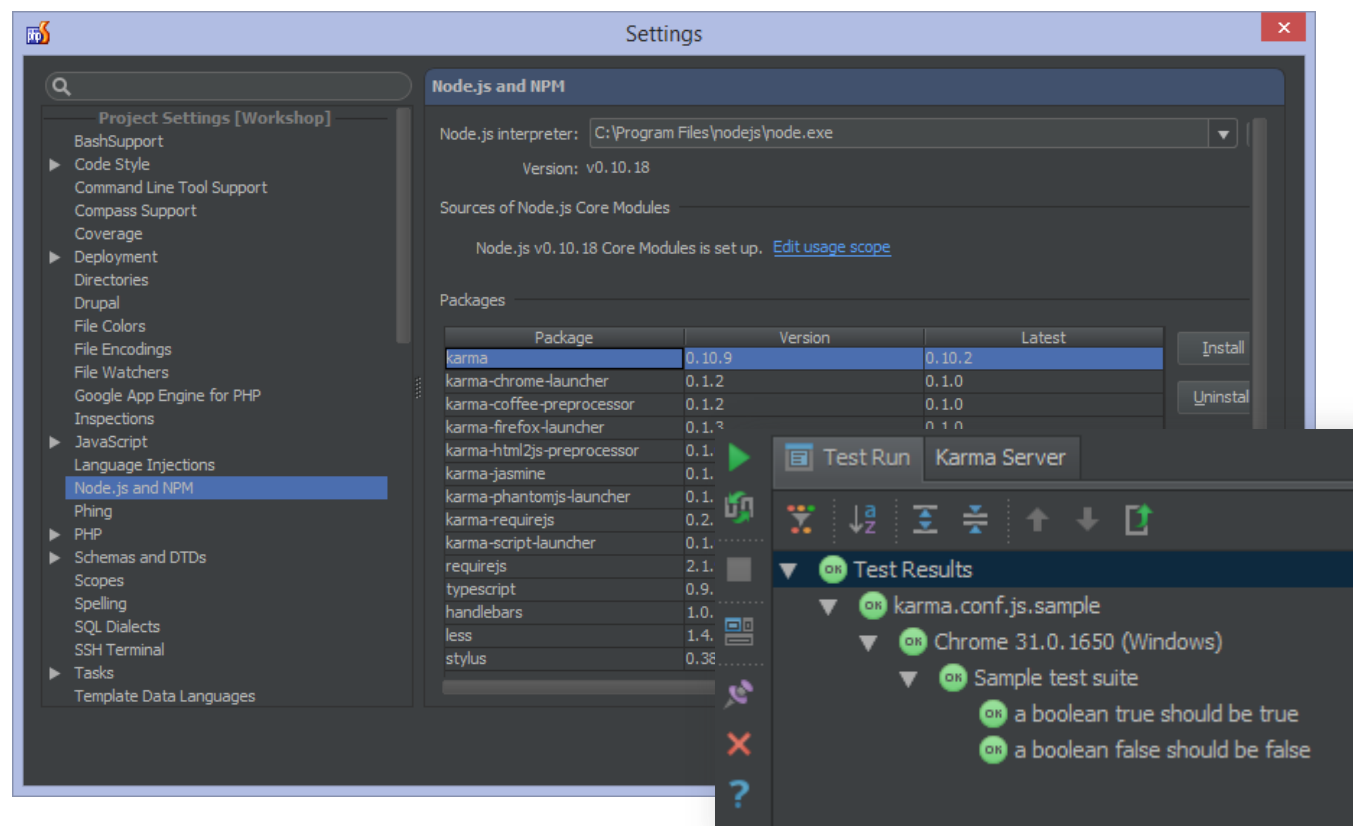
<https://www.jetbrains.com/phpstorm/webhelp/preparing-to-use-karma-test-runner.html>

[Installing JSTestDriver](#)

<https://www.jetbrains.com/phpstorm/webhelp/preparing-to-use-jstestdriver-test-runner.html>

Support for

- JSTestDriver Assertion framework
- Jasmine
- QUnit
- Mocha



Version Control

Version Control

Keep a log of all changes

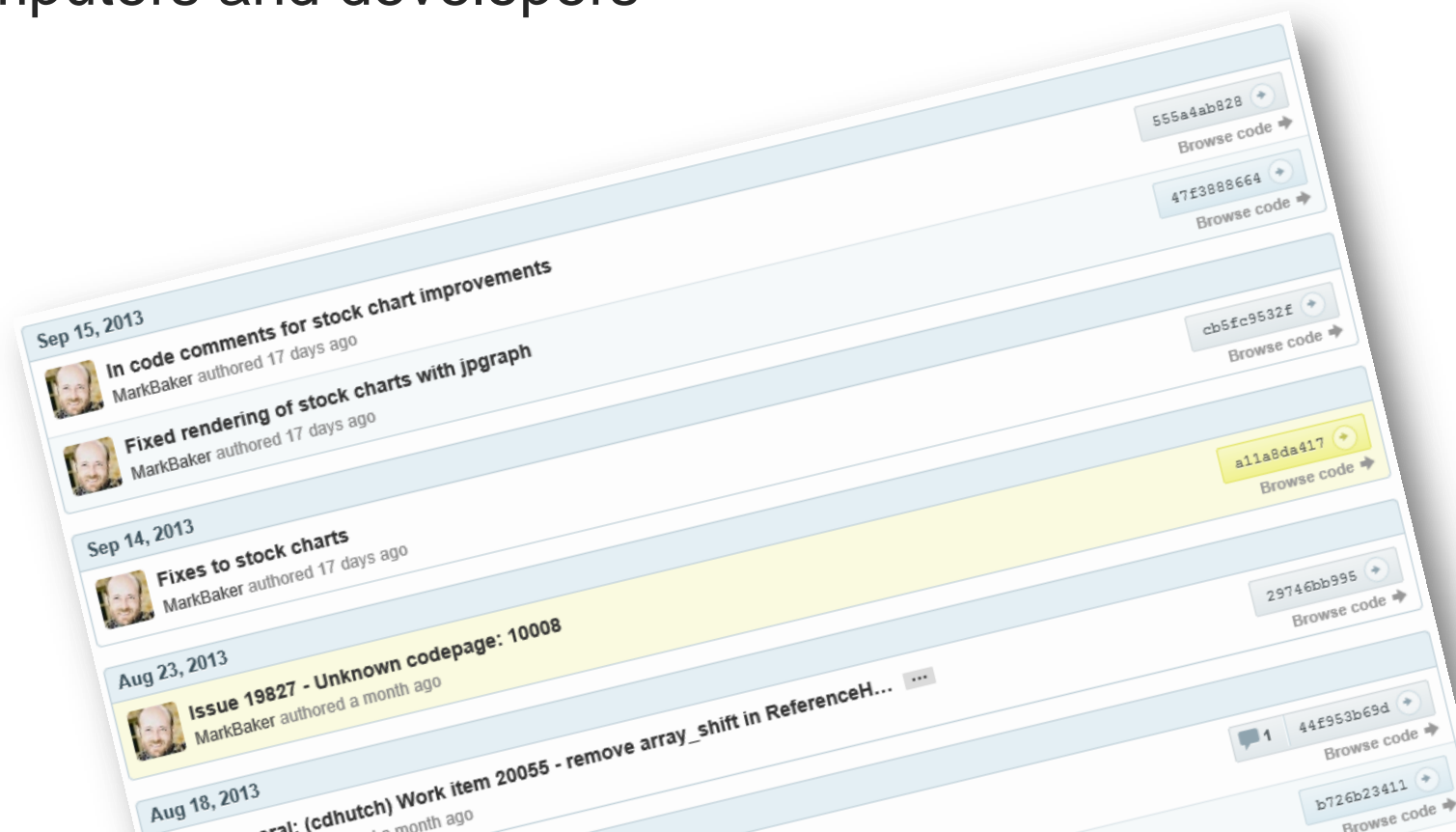
Synchronize copies across computers and developers

Go back in time

Who did what when why?

Compare versions

Rollback versions



Concepts

Repository (local / remote)

Checking out / cloning

Fetch copy from repository

Commit

Submit changes to repository

Revision / changeset

What changed? When? Who? Why?

Stored in order

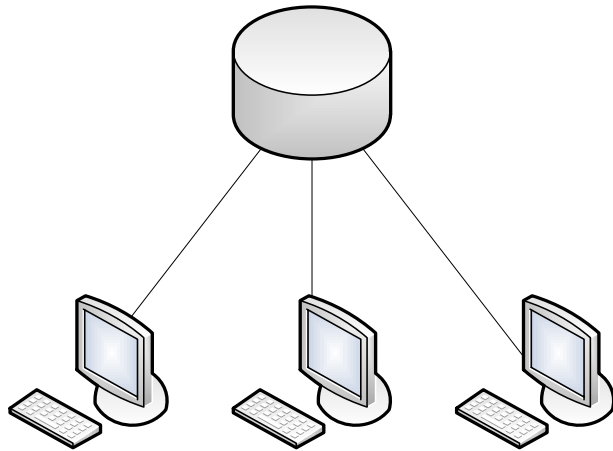
Conflicts

Diff

Branching / merging

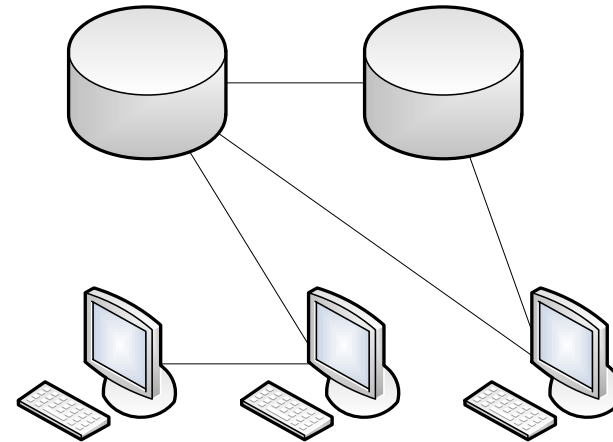
Flavours

CENTRALIZED VERSION CONTROL



Central repository = entire history
Devs work on a version and sync

DECENTRALIZED VERSION CONTROL



Devs have copy of entire history
Work on a version and sync with any

Products all use different terms...

Checkout / Clone / Export

Commit / Check In

Push / Submit

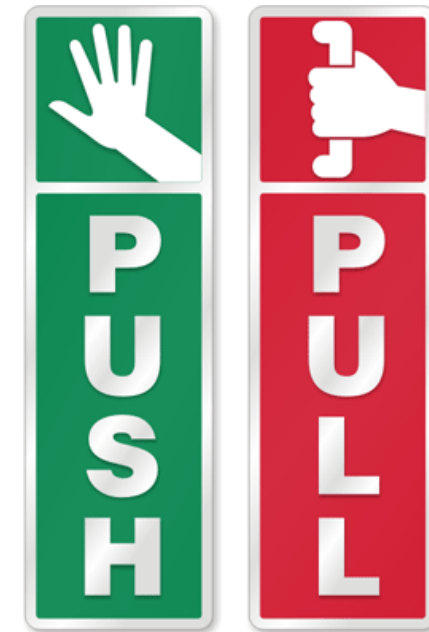
Revert / Roll back

History / Log

Annotate / Blame

Compare / Diff

...



But they are similar in nature!

Unified VCS functionality in the IDE

CVS

Git

Subversion

Mercurial

Perforce

TFS (via plugin)

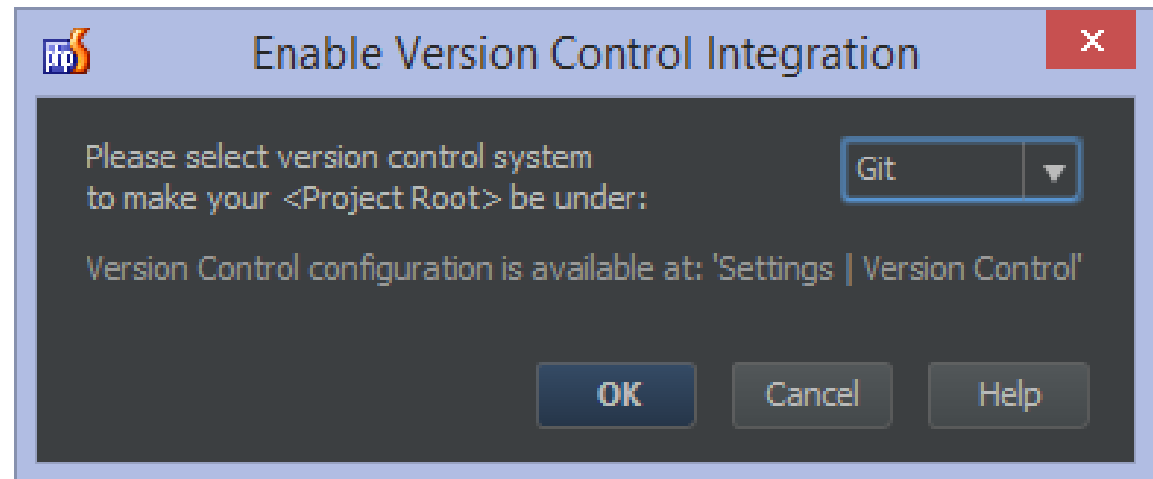
(others through plugins)

IDE enriched with specific functionality (e.g. push/pull)

Enabling VCS integration

Telling the IDE that the project is under a VCS system.

Use the VCS menu.



VCS operations

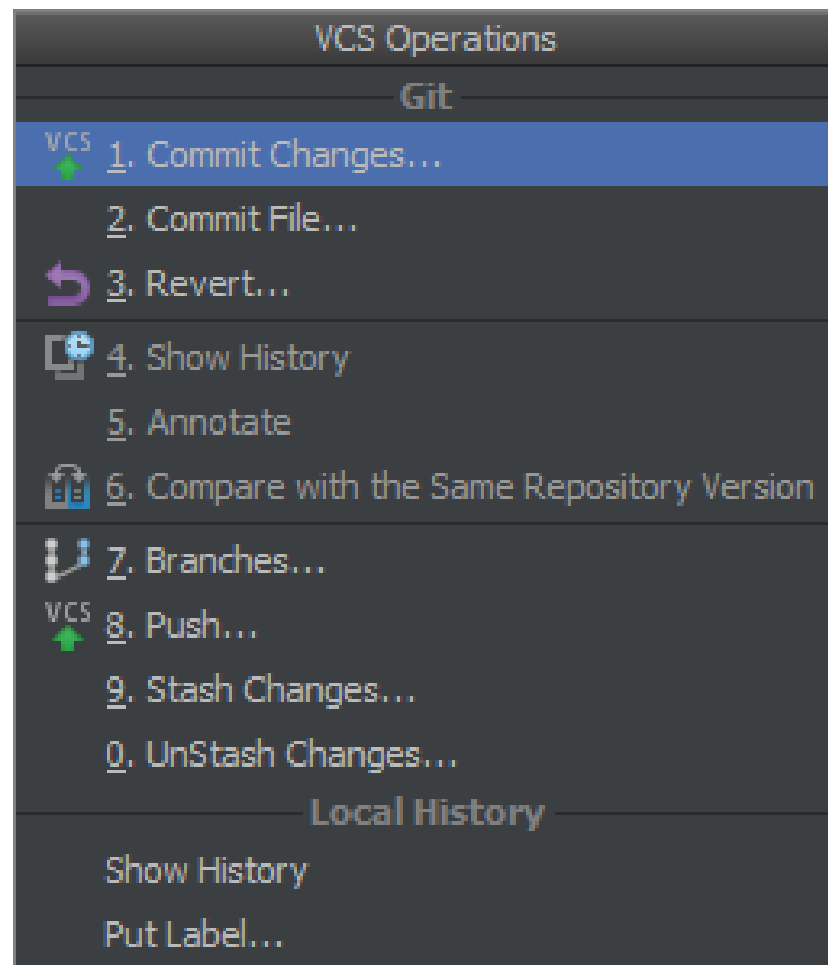
Perform operations on the version control system such as adding files, deleting files, committing changes and so on.



Alt+Backquote (Alt+`)



Ctrl+V



Committing Changes

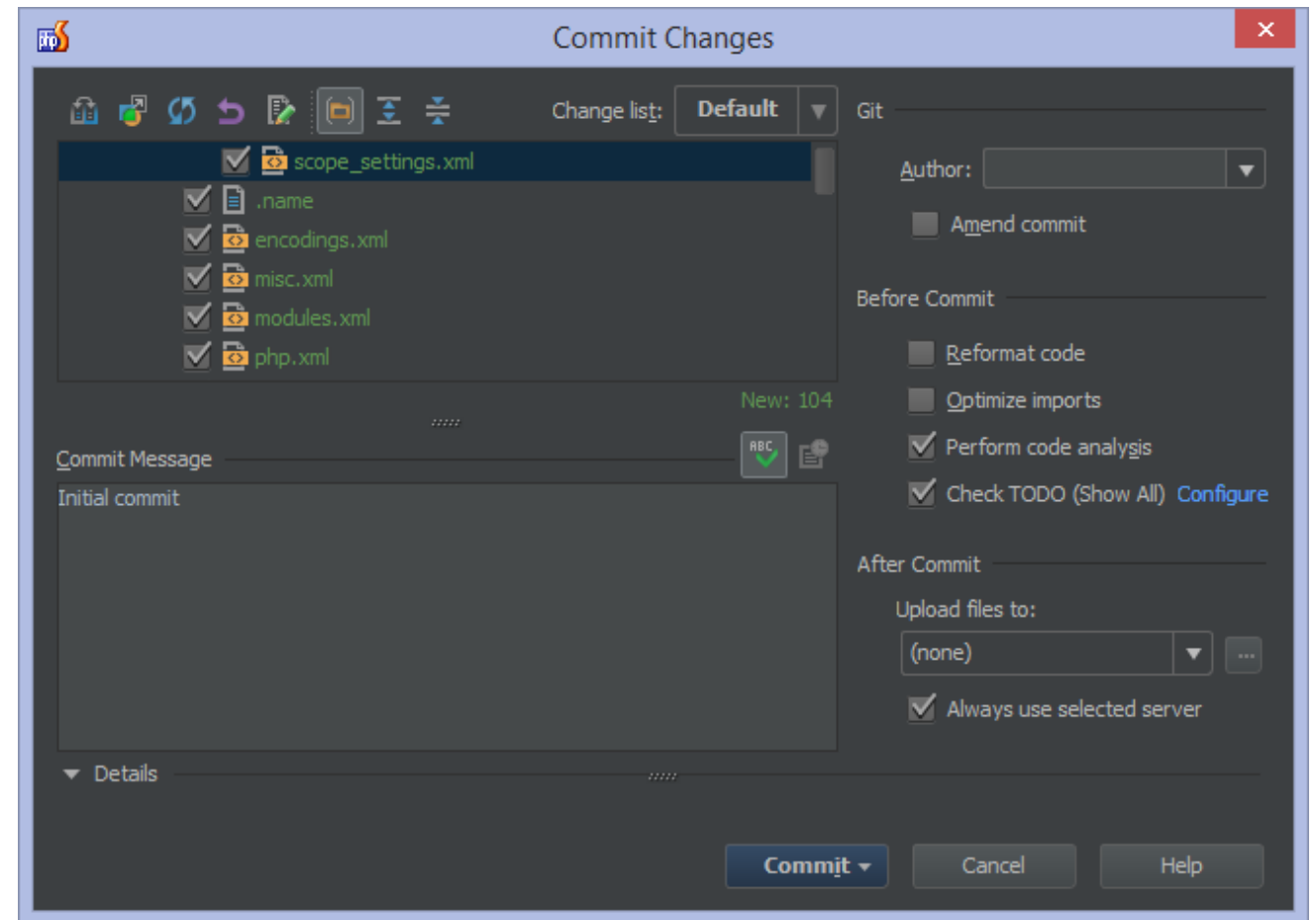
Persisting the current changeset as a logical operation in the VCS, with a comment, author, ...



Ctrl+K

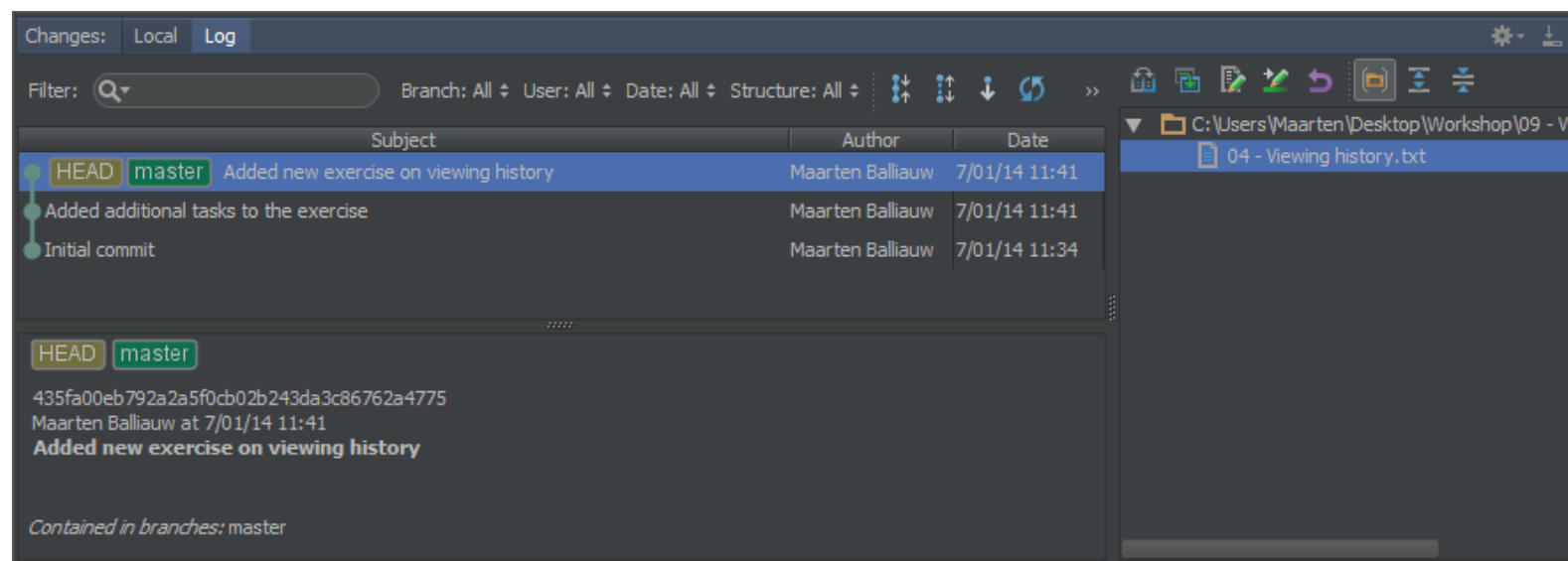


Command+K



Viewing History

View all information related to previous commits.



Alt+9



Command+9

Viewing Diff

Show changes made to a particular file.



Ctrl+D



Command+D

```
18b67a2f4e3673e8b2e1e00039fe11ddc9090249 (Read-only)
Committing Changes

Persisting the current changeset as a logical operation

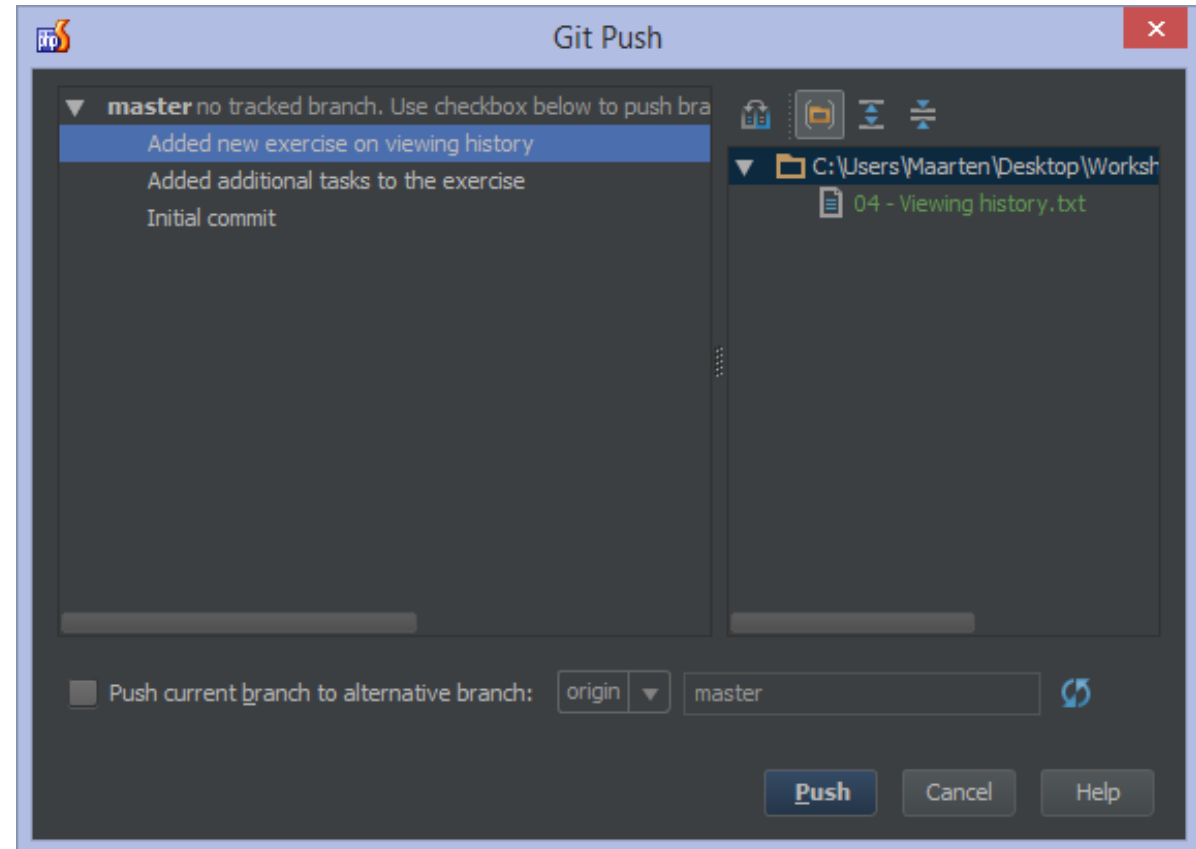
Ctrl+K (Windows/Linux)
Command+K (Mac OS X)

1. Commit the current changeset to the Git repository
2. The list on top shows changed, added and deleted files.
   Set the commit message to something describing the change.
   In this case this can be "Initial commit" but for a more
   descriptive message, use a number and a short description.
   Note that the IDE can automatically reformat code, which
   we can also upload changes to a remote server once we
   have committed the changes.
3. Commit the change.

c59a08aaa6bfc90ace6f05dfcc2a3caae96c0997 (Read-only)
1  Committing Changes
2
3  Persisting the current changeset as a logical operation
4
5  Ctrl+K (Windows/Linux)
6  Command+K (Mac OS X)
7
8
9  1. Commit the current changeset to the Git repository
10 2. The list on top shows changed, added and deleted files.
11   Set the commit message to something describing the change.
12   In this case this can be "Initial commit" but for a more
13   descriptive message, use a number and a short description.
14   Note that the IDE can automatically reformat code, which
15   we can also upload changes to a remote server once we
16   have committed the changes.
17 3. Commit the change.
18 4. The Project Tool Window will now show files in the
19   Project Tool Window, the file color also changes.
20 5. Start typing in the current file (enter your name in the
21   left gutter, denoting a change in the list).
22 6. Add a line of text below. Note that the editor
23 7. Commit all changes again and give the commit a more
```

Git/Mercurial: Push/Pull

Persist local copy of VCS repository to remote copy.



Changelists

Work on multiple logical changes at the same time.

Name changelist.

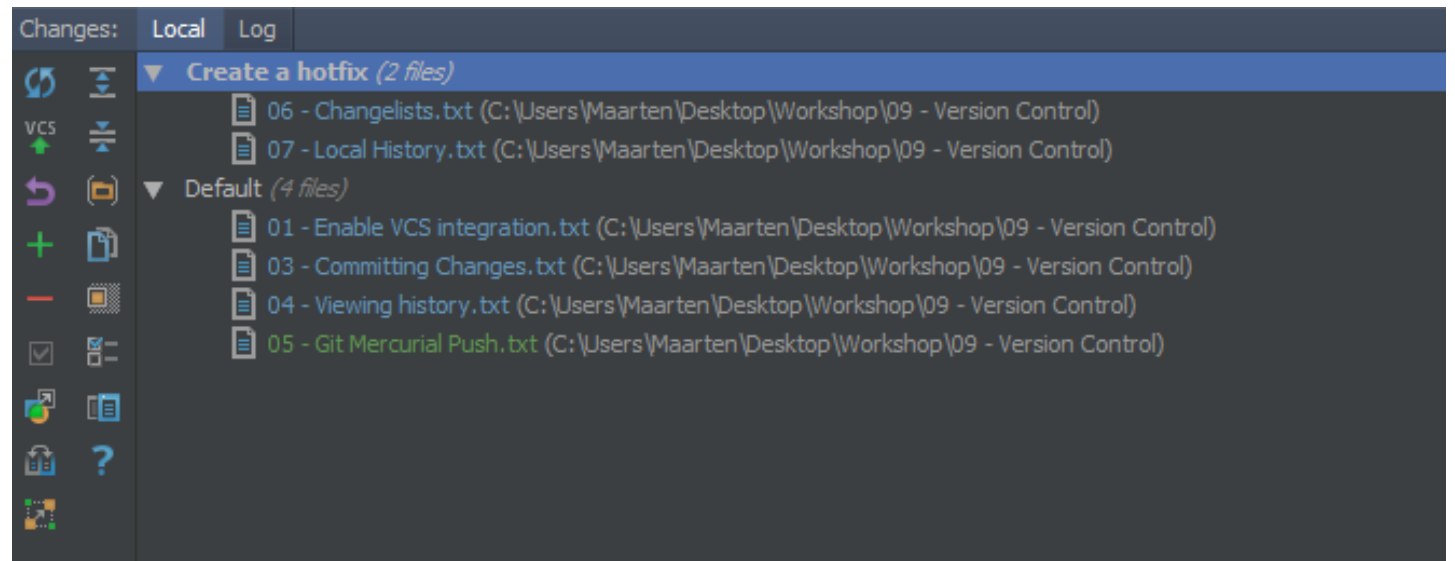
Can be committed / shelved.



Alt+9



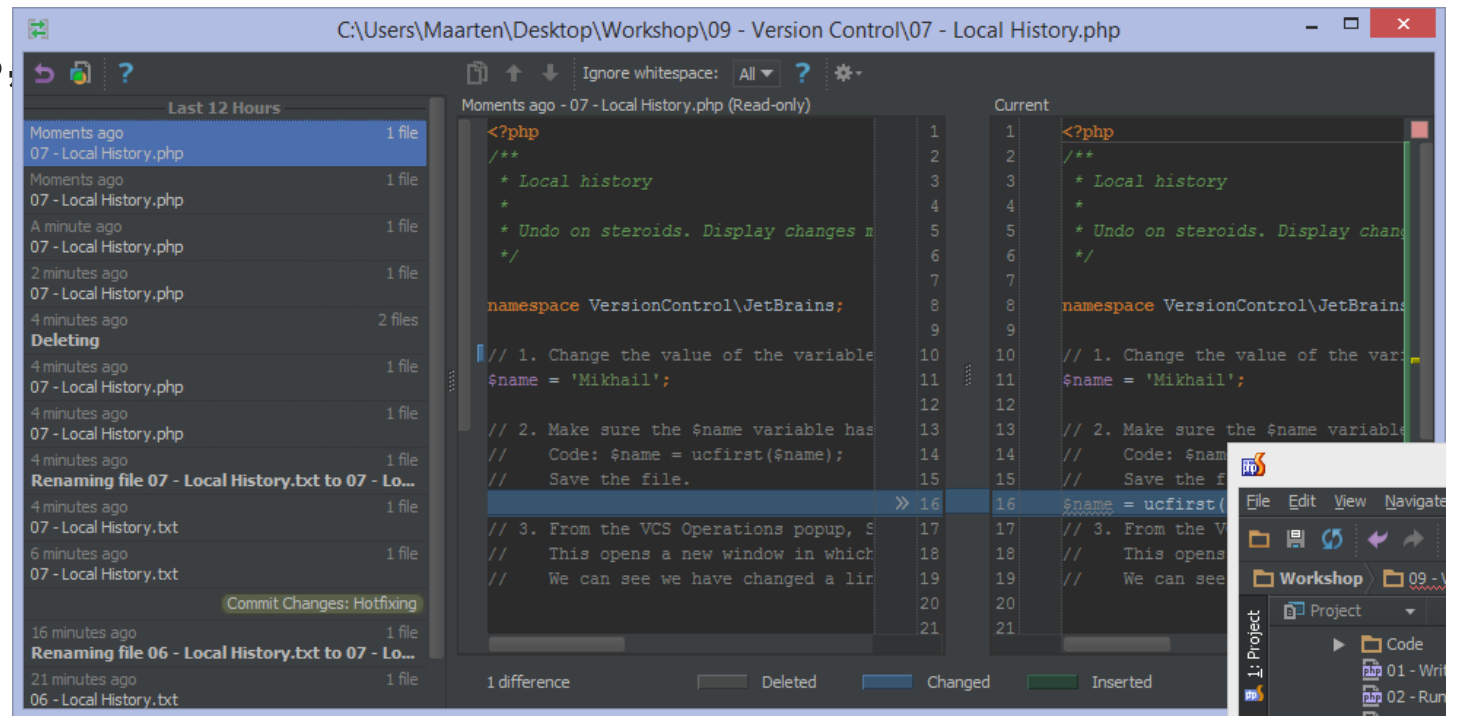
Command+9



Local history

Undo on steroids. Display changes made between commits, revert local changes.

...



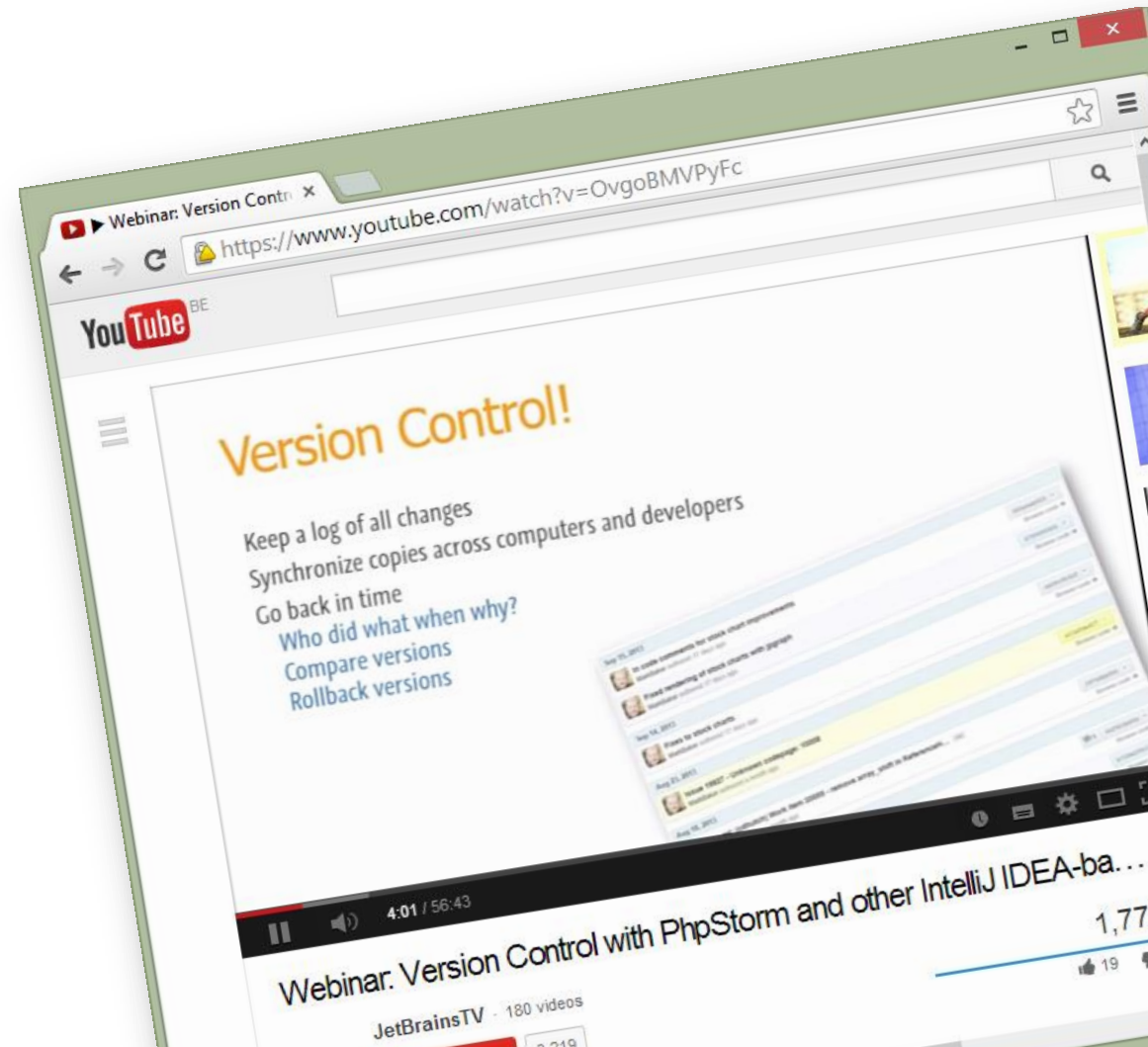
Resources

Webinar recording

<https://www.youtube.com/watch?v=OvgoBMVPyFc>

Web help

<http://www.jetbrains.com/phpstorm/webhelp/version-control-with-phpstorm-2.html>



Databases

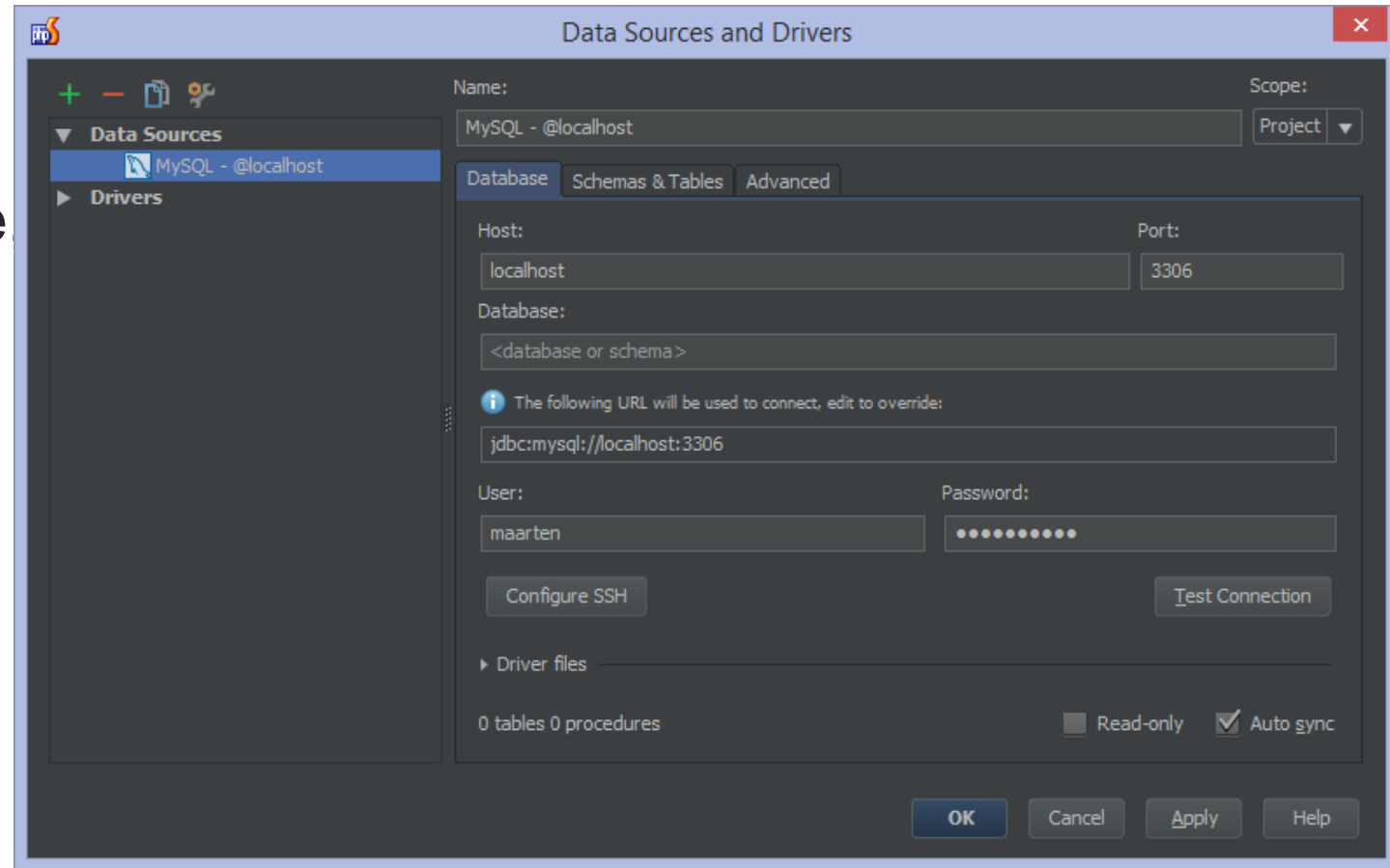
Connecting to a database (server)

Connecting to MySQL, DB2, Derby, SQL Server, Oracle, PostgreSQL, Sybase, H2, sqlite Google, ...

Specify connection details

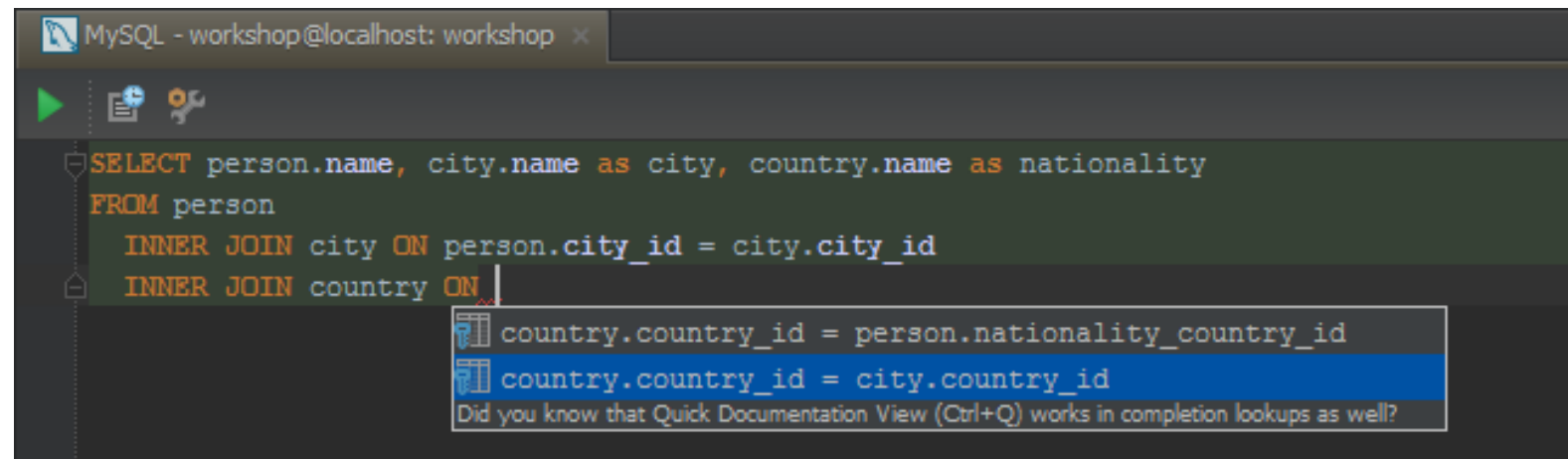
Download JDBC driver

Select schema / tables to work with



Database Console

SQL console with
autocompletion, inspections,
error highlighting,
autocompletion on joins, ...



The screenshot shows the MySQL console interface. The SQL code being entered is:

```
SELECT person.name, city.name as city, country.name as nationality
FROM person
INNER JOIN city ON person.city_id = city.city_id
INNER JOIN country ON
```

An autocomplete dropdown menu is visible, showing two suggestions:

- country.country_id = person.nationality_country_id
- country.country_id = city.country_id

Below the suggestions, a note reads: "Did you know that Quick Documentation View (Ctrl+Q) works in completion lookups as well?"



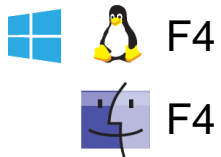
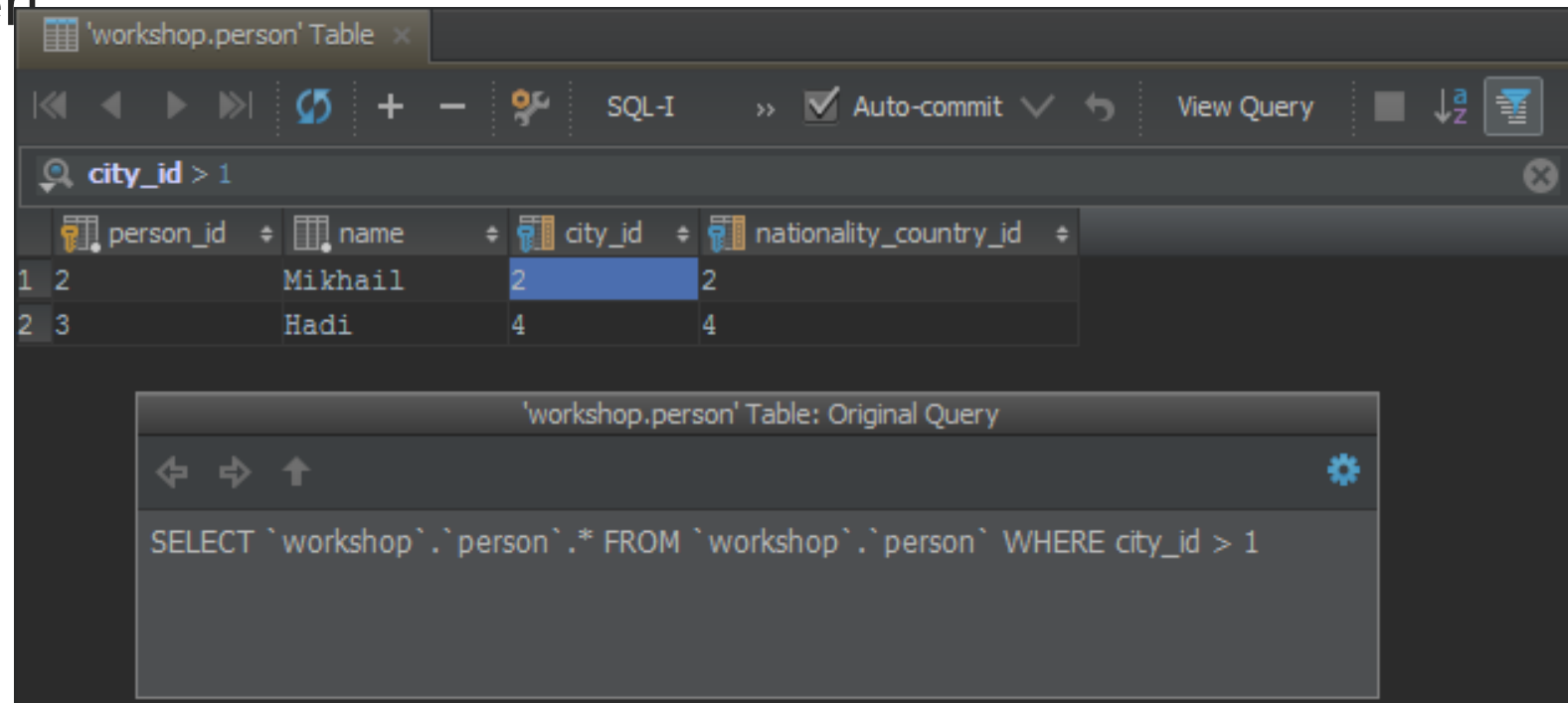
Ctrl+Shift+F10



Shift+Command+F10

Table Editor

Edit records in a table: insert, update, delete, ...



Create New Table

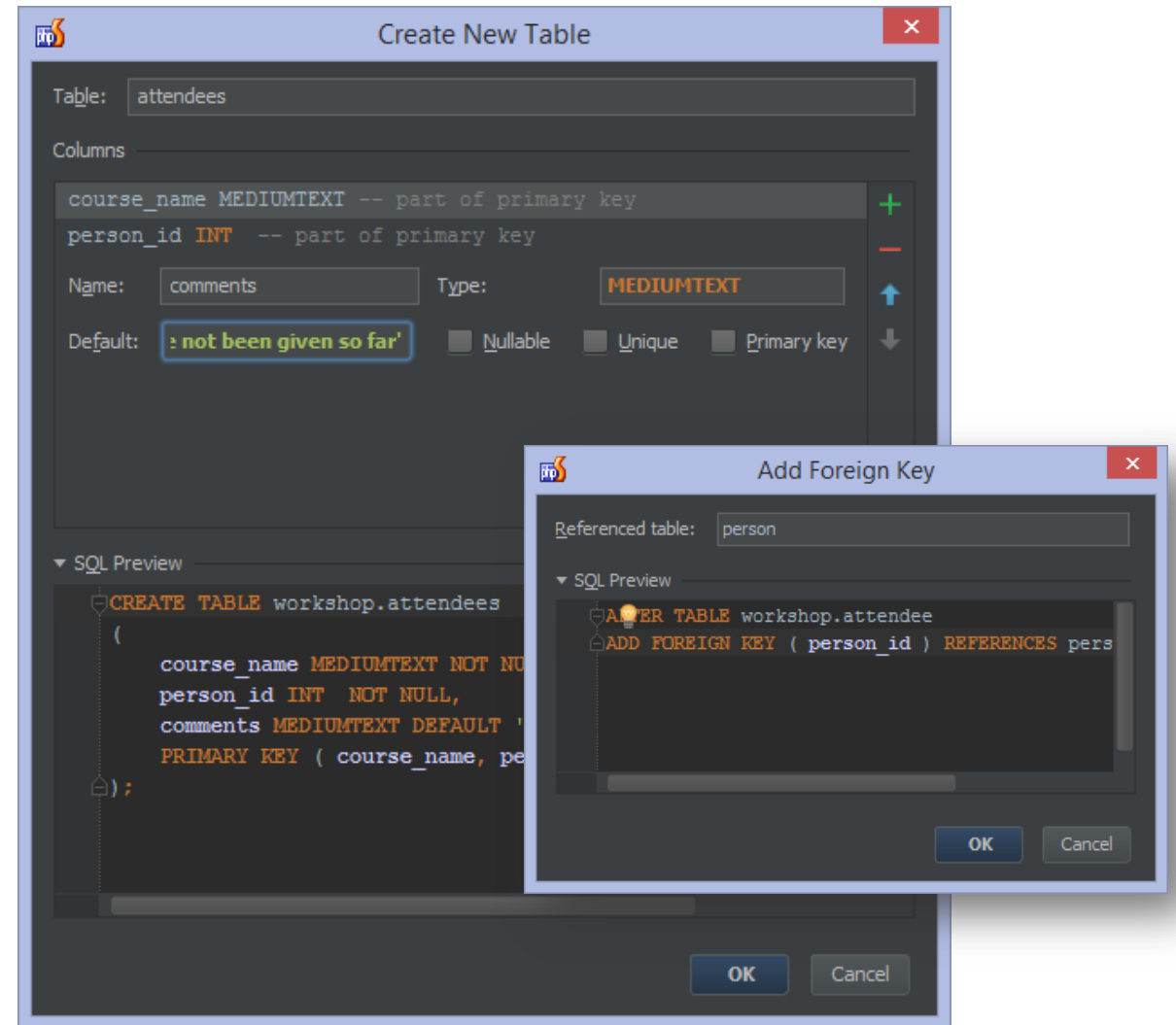
Create a new table in the database. Specify column options, default values, primary key, ...



Alt+Insert



Command+N



Database tools in PHP

Strings can be identified as a query, will provide autocompletion, highlighting, ability to go to query editor, run query right from PHP

```
$query = "SELECT person.na FROM person";
```

name	(people.person)	VARCHAR(45)
name	(workshop.person)	VARCHAR(45)
nationality_country_id	(people.person)	INT
nationality_country_id	(workshop.person)	INT

Press again to see global variants.



Ctrl+Space / Ctrl+Shift+Space



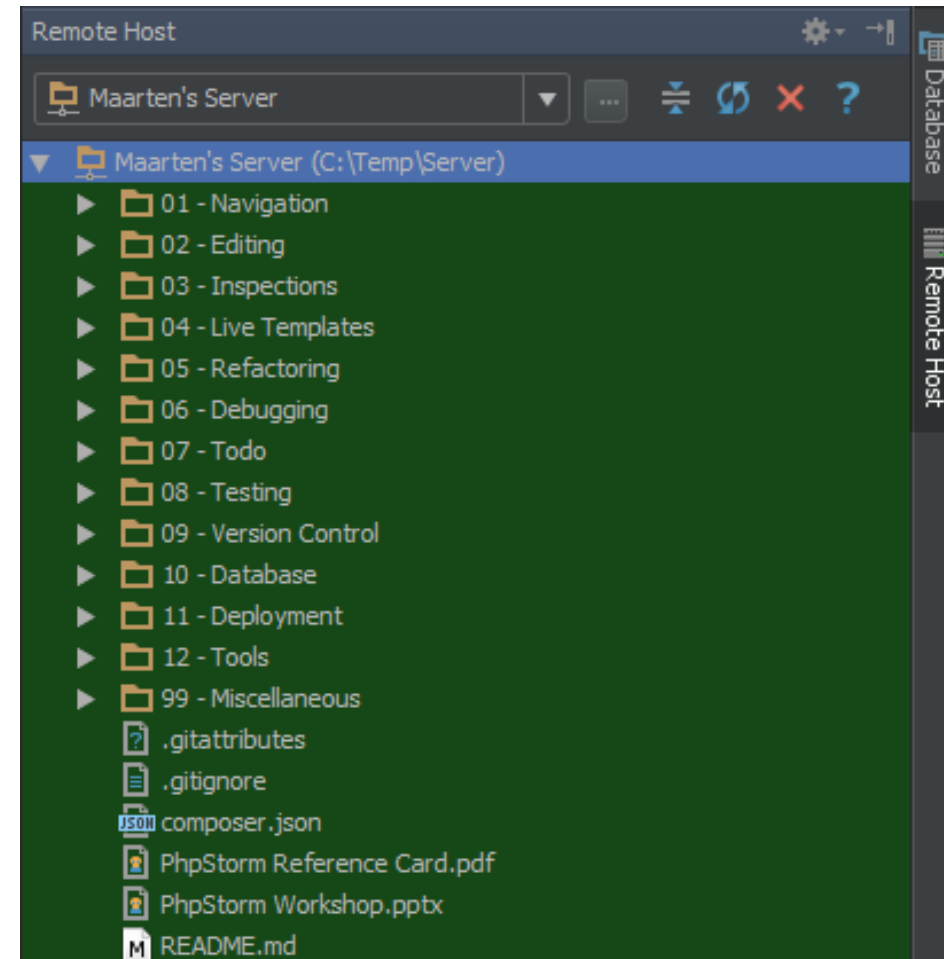
Ctrl+Space / Ctrl+Shift+Space

Deployment

Remote Hosts

Copy files and folders back and forth between local project and server.

Can be FTP, FTPS, SFTP, Local or Mounted Folder or In Place.

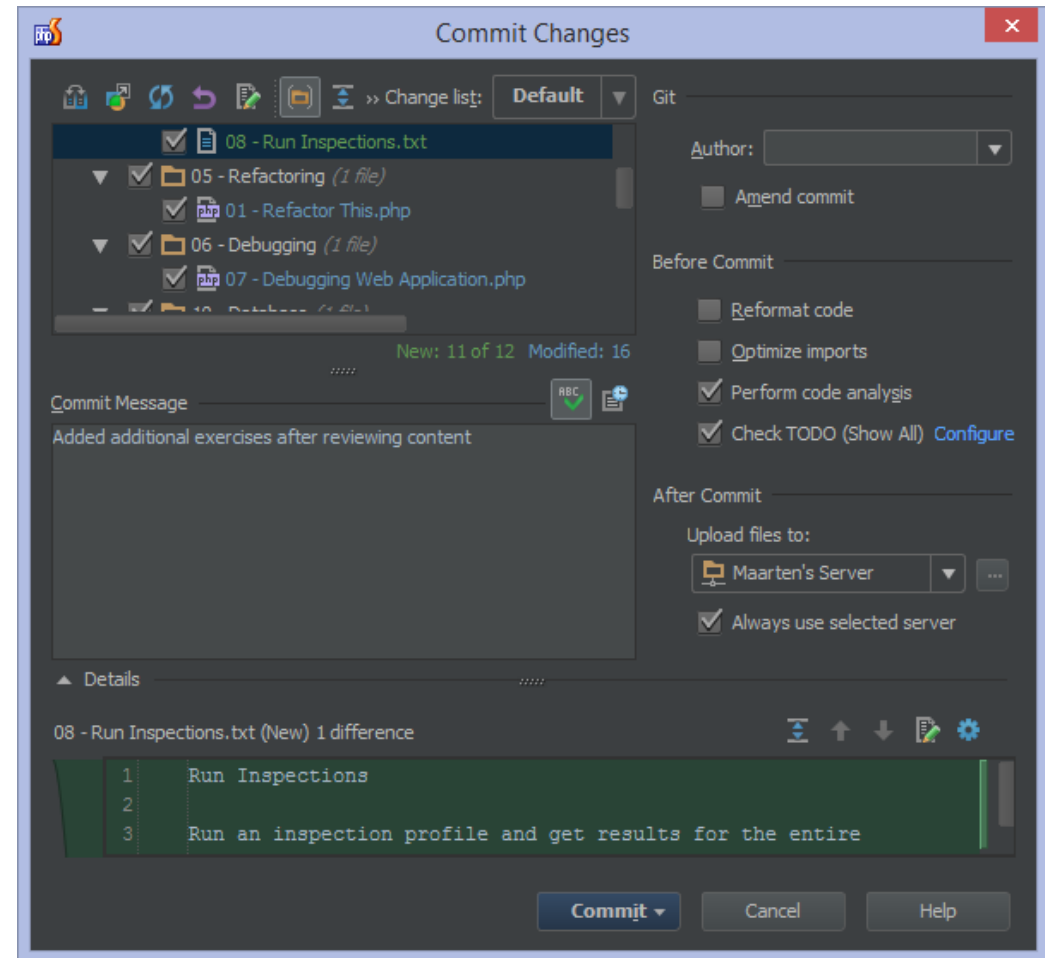


Integration with VCS

Upload files to server when committing to source control.

Useful for development servers.

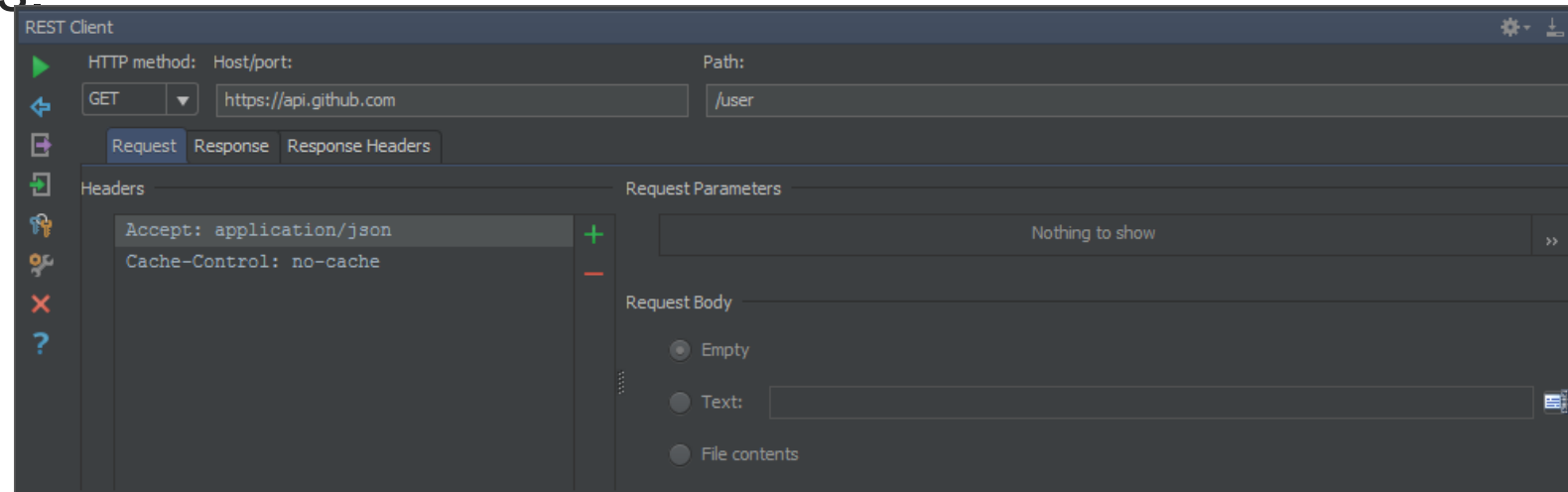
For production servers better to use continuous integration server.



Tools

REST Client

The built-in REST client allows testing web API's. Build requests, inspect responses.



Composer

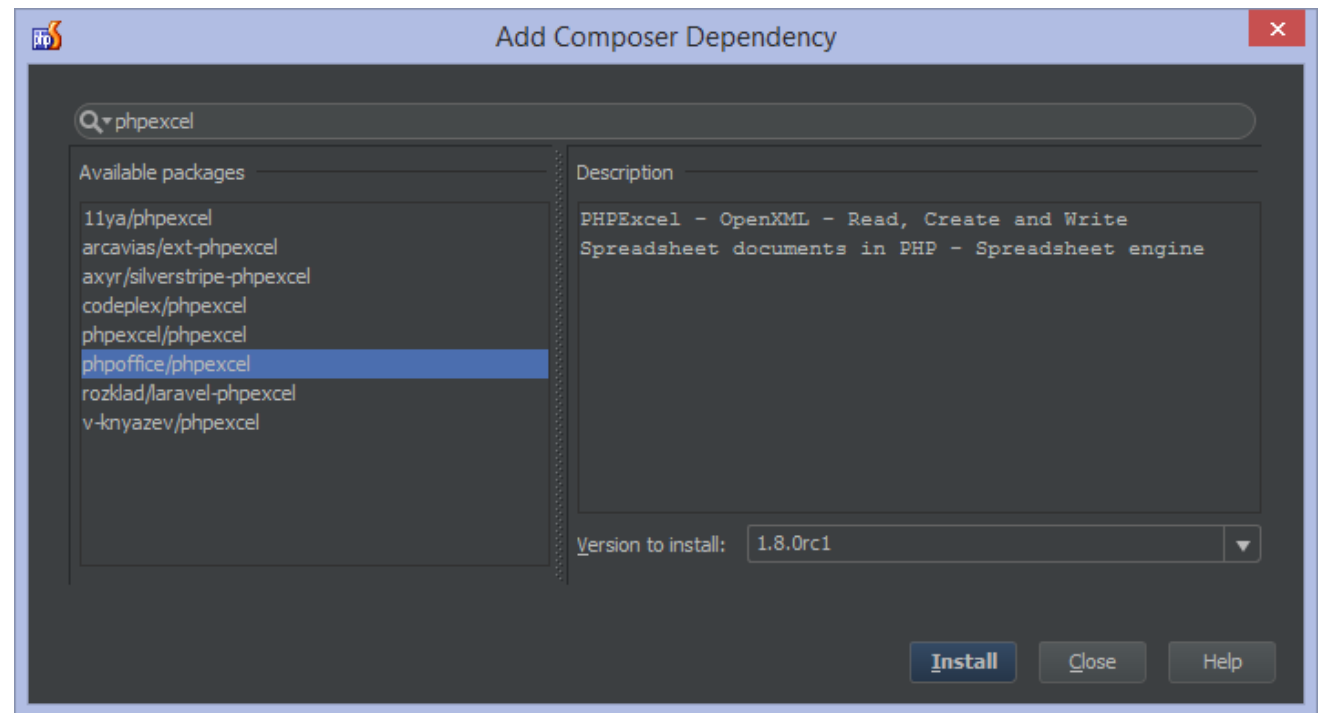
Work with Composer dependency manager in the IDE.

Initialize Composer

Search and install Composer dependencies

Create new project using Composer project type.

Command Line Tools (see further)



Command Line Tools

Invoke command line tools shipped with frameworks and libraries.

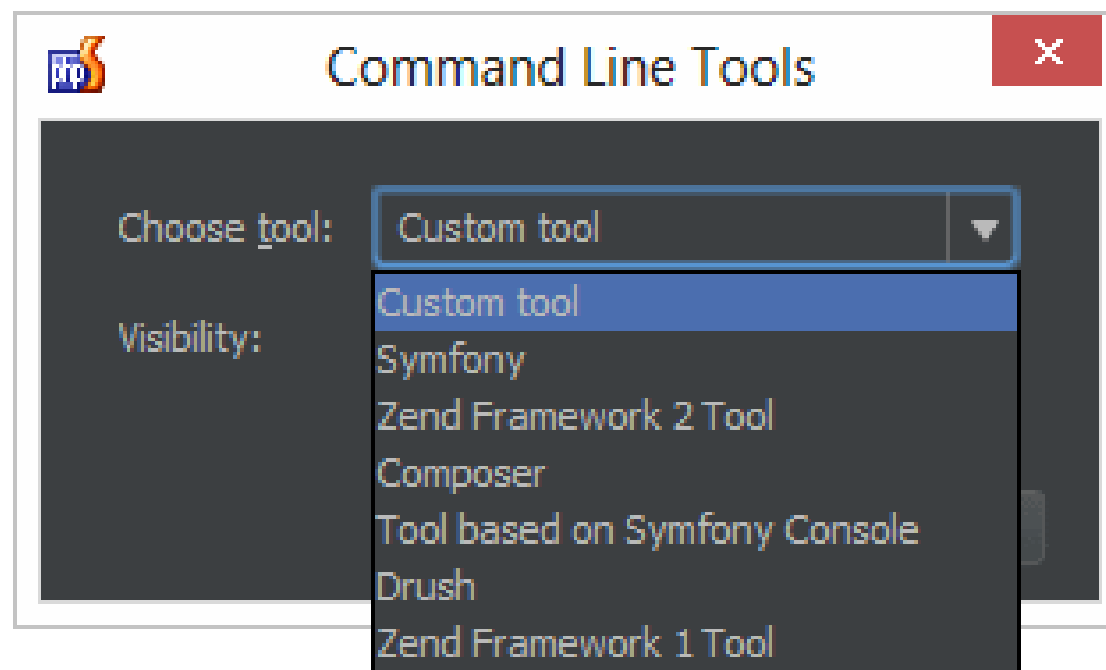
Composer, Zend Framework, Symfony, Symfony Console-based, Drush or roll your own

Autocompletion support

Not meant to be a full console/terminal!

  Ctrl+Shift+X

 Shift+Command+X

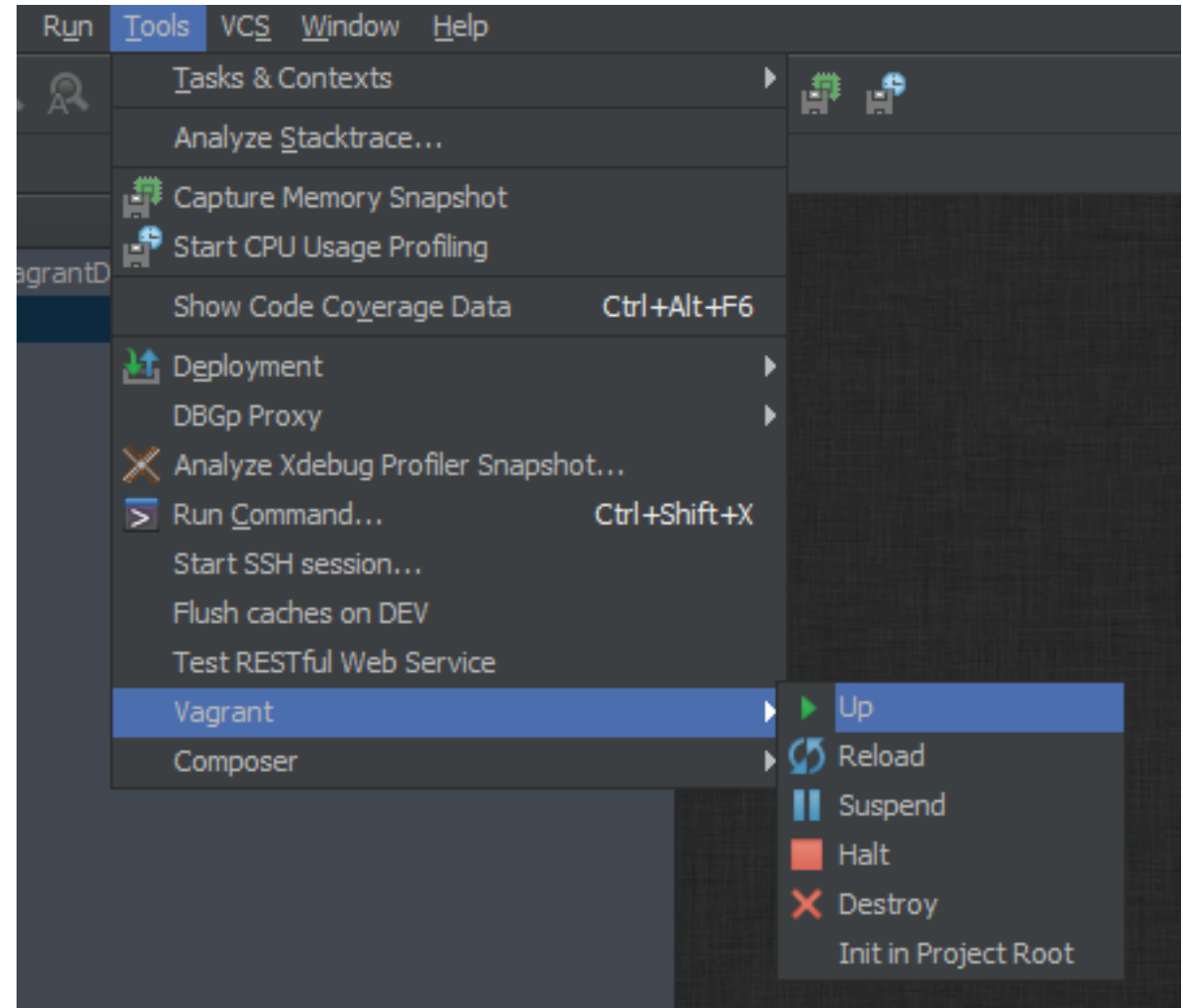


Vagrant

Vagrant is a tool which helps us create reproducible development environments by scripting a virtual machine.

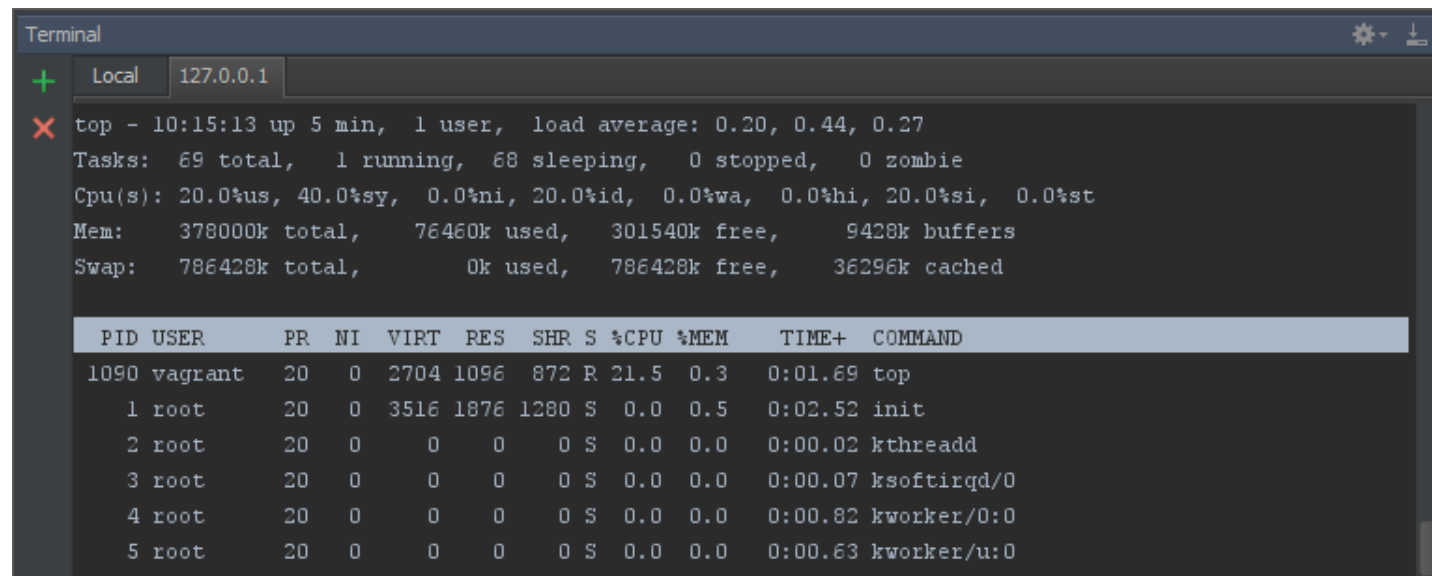
See

<http://blog.jetbrains.com/phpstorm/2013/08/vagrant-support-in-phpstorm/>



Remote SSH Terminal

Connect to a remote SSH server. Provides a full SSH terminal in the IDE.



The screenshot shows a terminal window titled "Terminal" with a tab labeled "Local 127.0.0.1". The terminal displays the output of the 'top' command, showing system statistics and a list of running processes. The processes listed include 'top', 'init', 'kthreadd', 'ksoftirqd/0', 'kworker/0:0', and 'kworker/u:0'.

```
top - 10:15:13 up 5 min,  1 user,  load average: 0.20, 0.44, 0.27
Tasks:  69 total,   1 running,  68 sleeping,   0 stopped,   0 zombie
Cpu(s): 20.0%us, 40.0%sy,  0.0%ni, 20.0%id,  0.0%wa,  0.0%hi, 20.0%si,  0.0%st
Mem:   378000k total,   76460k used,  301540k free,   9428k buffers
Swap:  786428k total,    0k used,   786428k free,   36296k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1090	vagrant	20	0	2704	1096	872	R	21.5	0.3	0:01.69	top
1	root	20	0	3516	1876	1280	S	0.0	0.5	0:02.52	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.82	kworker/0:0
5	root	20	0	0	0	0	S	0.0	0.0	0:00.63	kworker/u:0

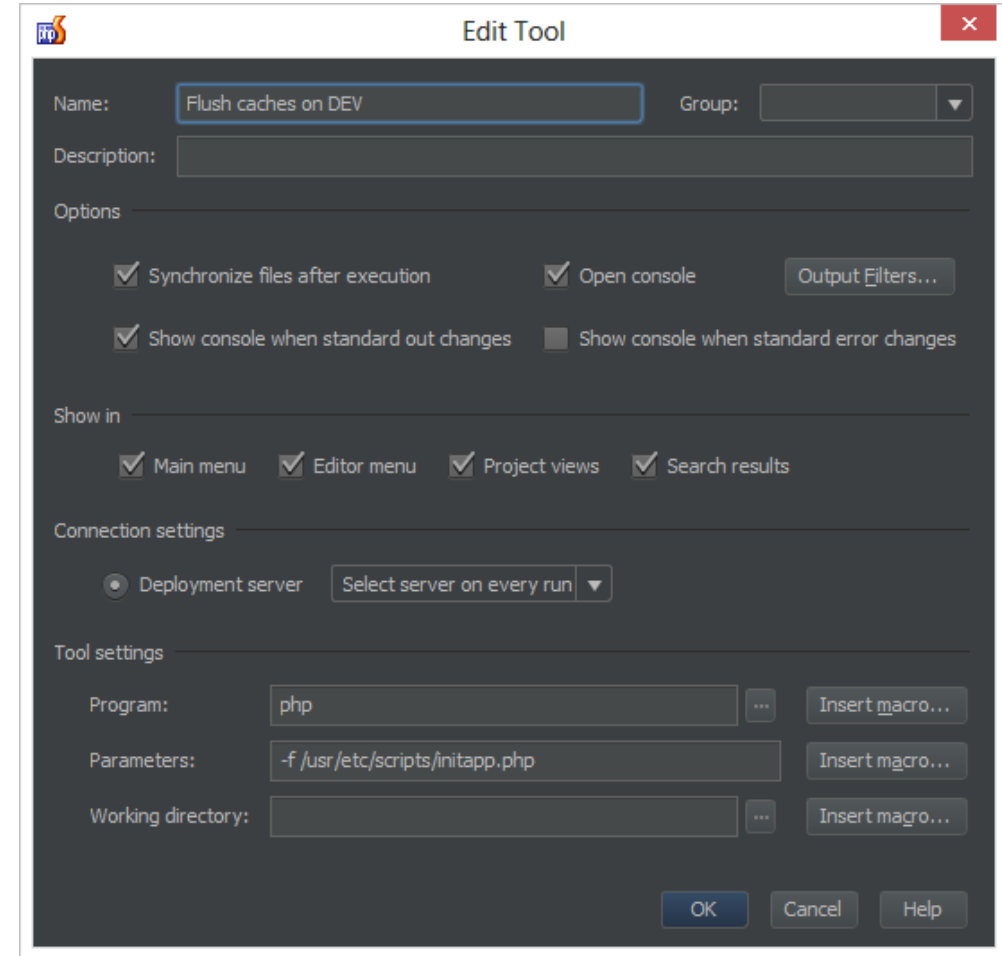
Remote SSH External Tools

Invoke remote commands over SSH without having to manually log in to the remote server.

Add them to PhpStorm menus for easy access.

Assign keyboard shortcut.

<http://confluence.jetbrains.com/display/PhpStorm/Using+the+PhpStorm+built-in+SSH+terminal+and+remote+SSH+external+tools#UsingthePhpStormbuilt-inSSHterminalandremoteSSHexternaltools-WorkingwithremoteSSHexternaltools>



Local Terminal

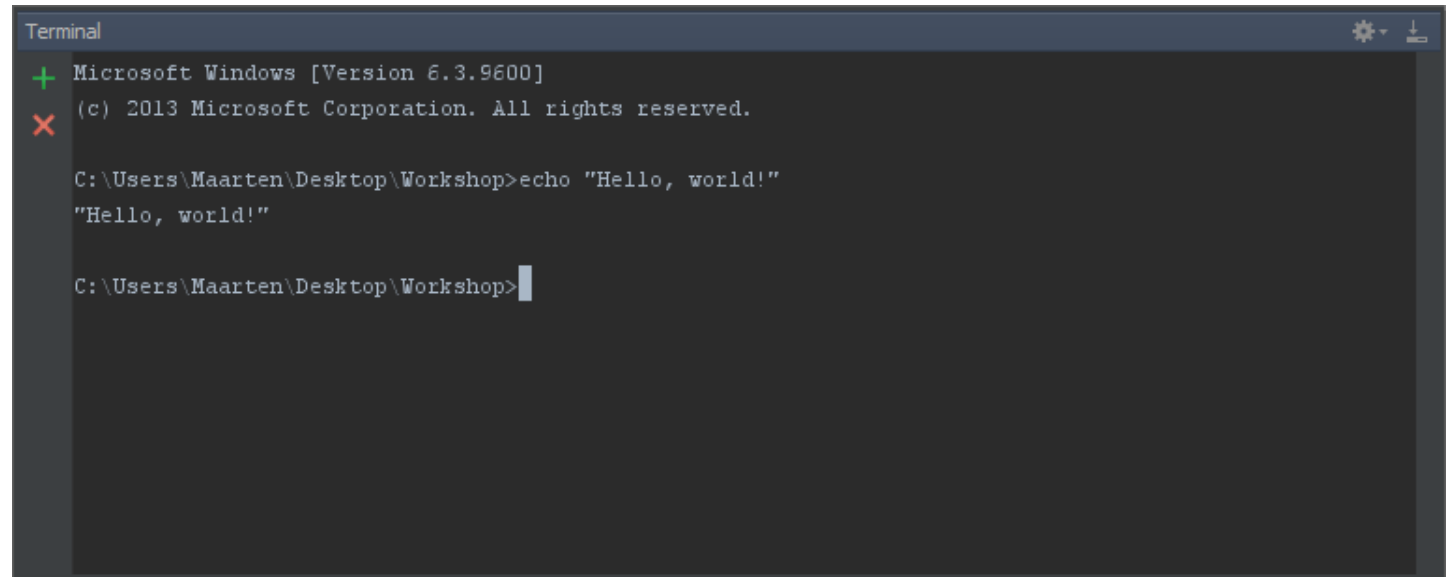
Provide a local terminal. Works on any platform supported by PhpStorm.



Ctrl+Space / Ctrl+Shift+Space



Ctrl+Space / Ctrl+Shift+Space



```
Terminal
+ Microsoft Windows [Version 6.3.9600]
+ (c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Maarten\Desktop\Workshop>echo "Hello, world!"
"Hello, world!"
C:\Users\Maarten\Desktop\Workshop>
```

Plugins

PhpStorm = IntelliJ IDEA – JAVA + plugins

Base IDE enriched with plugins

Some plugins are for paid IDE only

Which is cool: WebStorm features can be installed in PhpStorm

A lot of free / open source plugins at <http://plugins.jetbrains.com/?phpStorm>

Install from **IDE Settings | Plugins**



Some interesting JetBrains plugins

LiveEdit - view changes live in browser and vice-versa

NodeJS - install Node.js packages (NPM) in a similar way to Composer

IdeaVim - adds VIM capabilities to the IDE

Handlebars/Mustache and EJS - plugins for working with client-side template engines

JSTestDriver and Karma - run JavaScript unit tests

Puppet Support - provides editor support for puppet, very convenient with Vagrant

...



Some interesting open source plugins

'Copy' on steroids - copy rich text from the editor

Markdown - provides Markdown editor support and syntax highlighting

BashSupport - support bash files in editor

AngularJS - support for AngularJS

Symfony, Yii, CakeStorm, ... - framework-specific plugins

CodeGlance - the editor scrollbar on steroids

EditorConfig - support for EditorConfig settings

Get Gist - fetch a Gist from GitHub and insert it into code

...

There is much more...

What we did not cover...

Be sure to try these afterwards!

File Watchers – monitor a file and run command when it changes (e.g. LESS to CSS)

Google App Engine – work with GAE development environment and deploy

Drupal Support – project support, code style, hooks, ...

Live Edit – view changes live in browser and vice-versa

Issue Tracker – open and browse issues from YouTrack, JIRA, GitHub, ...

HTML, CSS, JS tools

Framework Integration – integrate with Symfony, Yii, ...

GitHub support – pull requests, gists, ...

Resources

Resources

Keyboard reference - <http://bit.ly/phpstorm-shortcuts>

Blog - <http://blog.jetbrains.com/phpstorm/>

Web Help - <http://www.jetbrains.com/phpstorm/webhelp/>

Tutorials - <http://confluence.jetbrains.com/display/PhpStorm/Tutorials>

Courseware (videos) – <http://bit.ly/phpstorm-videos>

Webinars - <http://blog.jetbrains.com/phpstorm/tag/webinar/>

Twitter - <https://twitter.com/phpstorm>

