iCampus Development

James Antrim, Mariusz Homeniuk, Mehmet Ali Pamukci,
Dennis Priefer, Wolf Rost, Niklas Simonis, Dennis Voigtländer

May 25, 2016

Contents

Chapter 1

Introduction

This documents explains how to set up and develop in the context of the iCampus Working Group. All instructions in this document are based upon the Windows 8.1 operating system.

Chapter 2

The Development Environment

In order to develop on a personal computer, must certain steps be performed in various systems, as well as various programs installed and configured. In this chapter these items will be discussed.

2.1 Local Web Server Installation and Configuration

In order for a locally developed project to be deployed, and to guarantee access to all required components a web server must be installed locally.

2.1.1 XAMPP

In the iCampus web development group we use XAMPP from Apache Friends for our local web servers. Apache Friends describes XAMPP as follows:

> *XAMPP is a completely free, easy to install Apache distribution containing MySQL, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.*[1]

As per the description the distributions of XAMPP available from Apache Friends come included with various useful tools which are also used within iCampus, such as PHP (server-side scripting language)abd MySQL (database management system), as well as many other tools which are optional for iCampus such as FileZilla (FTP Server) and Mercury (Mail Server).

Download and install the latest XAMPP installation. Keeping in mind that at least MySQL and phpMyAdmin need to be installed to function in the iCampus environment.

[1] https://www.apachefriends.org/index.html September 19, 2014

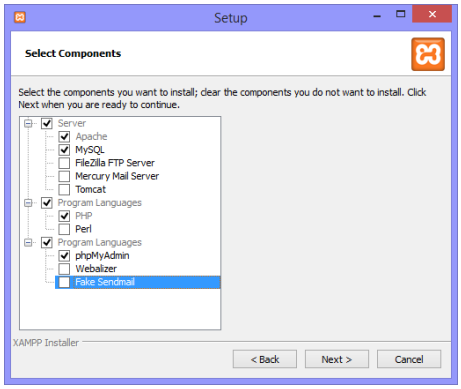

Figure 2.1: XAMPP Minimum Installation

2.1.2 PHP Configuration

The next step is the configuration of PHP for error reporting using xdebug. To do this open your php.ini file using the XAMPP control panel or the file explorer. This file is typically located at `C:/xampp/php/php.ini`.

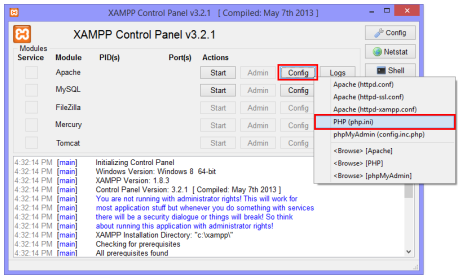

Figure 2.2: XAMPP Control Panel

The following modifications should then be made to the file:

```
// Reports all PHP Errors, Warnings, and Breaches of PHP Standards
error_reporting = E_ALL | E_STRICT

// Turns off error buffering => errors reported immediately, necessary for the debugger
output_buffering = Off
```

```
// Debugger Settings (may only need to be commented in)
[XDebug]
zend_extension="C:\xampp\php\ext\php_xdebug.dll"
xdebug.remote_enable=true
xdebug.remote_host=localhost
xdebug.remote_port=9000
xdebug.remote_handler=dbgp
xdebug.profiler_enable=1
xdebug.profiler_output_dir="C:\xampp\tmp"
```

The interface shown in Figure 2.2 above will also later used to start and stop the server service itself, Apache, as well as the the supporting database management service, MySQL, by pressing the corresponding "Start"/"Stop" button.

2.1.3 PEAR Configuration

In order to eliminate possible error sources, PEAR must next be configured. To this end run cmd.exe as an administrator and execute the following commands.

```
// Navigate to the PHP Directory
cd C:\xampp\php

// Display the PEAR Configuration
pear config-show
```

Directory Configuration

The following settings which may refer to C:\php\pear need to be changed to point to C:\xampp\php\pear should they not already do so.

```
pear config-set doc_dir C:\xampp\php\pear\docs
pear config-set cfg_dir C:\xampp\php\pear\cfg
pear config-set data_dir C:\xampp\php\pear\data
pear config-set cache_dir C:\xampp\php\pear\cache
pear config-set download_dir C:\xampp\php\pear\download
pear config-set temp_dir C:\xampp\php\pear\temp
pear config-set test_dir C:\xampp\php\pear\tests
pear config-set www_dir C:\xampp\php\pear\www
pear config-set php_ini C:\xampp\php\php.ini
```

Channel Configuration

Next PEAR itself and its channel list needs to be updated, so that later updates for extensions can be automatically pulled from their respective channels. Thereafter the auto_discover variable is set to on, this allows new channels to be discovered and dependencies to be resolved automatically.

```
// Ensures the latest PEAR version
pear channel-update pear.php.net

// Empties the PEAR Cache
pear clear-cache

// Turns on Automatic Discovery
pear config-set auto_discover 1

// Update the Channel list
pear update-channels

// Uninstalls CodeSniffer in Case a Deprecated Version is Present
pear uninstall PHP_CodeSniffer

// Installs PHP CodeSniffer
pear install PHP_CodeSniffer-1.5.6

// Discovers the PHP Mess Detector channel
pear channel-discover pear.phpmd.org

// Lists available packages
pear remote-list -c phpmd

// Installs PHP Mess Detector Package 1.5.0, actual as of 27 Sept, 2014
pear install phpmd/PHP_PMD-1.5.0
```

More about Code Sniffer and Mess Detector in Section 2.4.

2.1.4 PHPUnit

The following text describes the installation of PHPUnit.[2]

For Linux Users

```
wget https://phar.phpunit.de/phpunit.phar
chmod +x phpunit.phar
sudo mv phpunit.phar /usr/local/bin/phpunit
```

For Windows Users

1. Create a directory for PHP binaries; e.g., `C:\bin`.
2. Add your PHP directory `C:\xampp\php` and the above created directory to your system's PATH environment variable. See appendix **??**.
3. Download https://phar.phpunit.de/phpunit.phar and save the file as C:\bin\phpunit.phar
4. Open a command line (e.g., press Windows+R > type cmd > ENTER)
5. Create a wrapping batch script (results in C:\bin\phpunit.cmd):

```
C:\Users\username> cd C:\bin
C:\bin> echo @php "%~dp0phpunit.phar" %* > phpunit.cmd
```

Open a new command line and confirm that you can execute PHPUnit from any path and it is the right PHPUnit version you installed previous:

```
> phpunit --version
> PHPUnit x.y.z by Sebastian Bergmann.
```

[2] For the original documentation see https://phpunit.de/manual/current/en/installation.html

2.2 Git

For repository and versioning iCampus uses Git. To this end Git must first be downloaded from Git, installed and configured.

2.2.1 Installation

Command Context

Part of the configuration is deciding which context you would like to be able to use Git in. In iCampus the second is used because it allows the developer later to use PHPStorm's own terminal to execute Git's commands, which eliminates the necessity for navigation to projects in Git Bash or Windows shell. More on PHPStorm in Section 2.3.

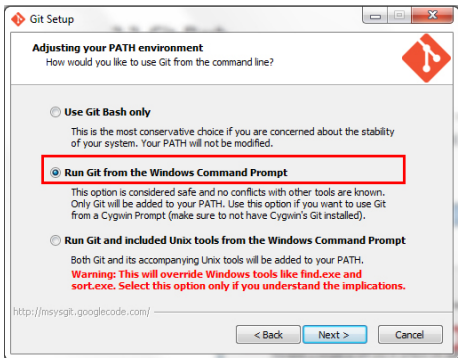

Figure 2.3: Command Context

Line Endings

The next screen lets one decide how file endings are pulled and committed. The iCampus Coding standards demand the Unix-style line endings. To this end only the second option is sufficient, because once style checking is enabled in your IDE the first and third options will, dependent on your system settings, report every single line of code as a breach of style, which removes any transparency whatsoever when searching for other errors.

2.2.2 User Configuration

In order to take part in the development process you must take several steps to configure your user account.

Global User Configuration

In Git Bash the global user name and email must be set so that repository actions can be associated with a user.

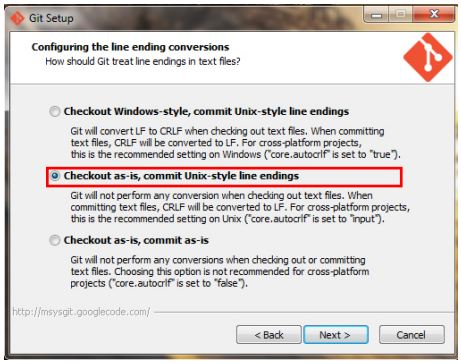

Figure 2.4: Line Endings

Start Git Bash and enter the following commands, replacing <First Name>, <Last Name>, and <Department> with your first and last names and the abbreviation of the department in which you are matriculated.

```
git config --global user.name "<First Name> <Last Name>"
git config --global user.email "<First Name>.<Last Name>@<Department>.thm.de"
```

You can confirm that the entries were saved correctly by entering the following command and comparing the associated values.

```
git config -l
```

SSH-Key Creation

After the user configuration is completed, an SSH-Key can be created using the following command in Git Bash.

```
ssh-keygen
```

The first question determines where the generated key will be saved and what its name will be. Per default this will be `C:/Users/<Windows Account Name>/.ssh/id_rsa`. This directory is important for the next steps, as well as later when repositories are cloned. Press enter to confirm the default. The next two questions will be to enter and confirm your password. You can also elect to have no password by pressing enter in answer to both questions.

2.3 PhpStorm

PhpStorm is the IDE used in iCampus. While others are allowed the use of a single IDE in the working group allows for the exchange of information and tips on how to get the most out of a single IDE. This makes everyone more productive and allows for a collaborative effort to solve any problems which may be IDE related.

PhpStorm was chosen for its ease of use and integration of many tools which simplify development, including CodeSniffer, database connection tools, SASS support, Git support, and many others. PhpStorm can be downloaded from the JetBrains Website.

2.3.1 Installation

During the installation process you will be asked which file extensions should be associated with PhpStorm. As we use PhpStorm for the development with all given extensions, all can be selected. As of PhpStorm 8 this also allows for the editing of single files without the creation of a project.

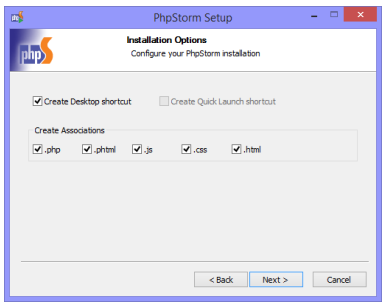

Figure 2.5: File Associations

For the most part the rest of the installation requires nothing special, however at the end you will be asked to chose between several themes and coloring/font style options. As this interface offers no preview it is best to go with the default settings. These selections can later be changed while running the PhpStorm which will actually show you what effects your selection has on its appearance.

After the installation starting PhpStorm will request the lisence information. Please register at `https://www.jetbrains.com/shop/eform/students` as a student. After this procedure you can activate PhpStorm using your JetBrains account.

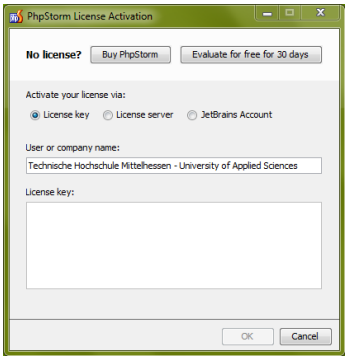

Figure 2.6: PhpStorm License Activation

2.4 Code Sniffer and Mess Detector

Now that we have checked out `lib_thm_core`, we can use the coding standards within it to set up PHP CodeSniffer.

> PHP_CodeSniffer is a PHP5 script that tokenises and "sniffs" PHP, JavaScript and CSS files to detect violations of a defined coding standard. It is an essential development tool that ensures your code remains clean and consistent. It can also help prevent some common semantic errors made by developers.[3]

PHP Mess Detector on the other hand searches for problems more abstract and more serious such as[4]:

- Possible bugs
- Suboptimal code
- Overcomplicated expressions
- Unused parameters, methods, properties

[3] http://pear.php.net/manual/en/package.php.php-codesniffer.intro.php Stand September 22, 2014
[4] http://phpmd.org/ Stand September 27, 2014

2.4.1 Standards Directory Inclusion

First we will create symbolic links from the Joomla and THMJoomla folders to the folder which holds the existing standards. Open the windows shell as administrator and enter the following in the form `Command Link Target`.

Joomla Standard

Command `mklink /d`

Link `C:\xampp\php\pear\PHP\CodeSniffer\Standards\Joomla`

Target `C:\<System Specific Path>\lib_thm_core\coding_standards\Joomla`

iCampus Standard

Command `mklink /d`

Link `C:\xampp\php\pear\PHP\CodeSniffer\Standards\THMJoomla`

Target `C:\<System Specific Path>\lib_thm_core\coding_standards\THMJoomla`

2.4.2 PhpStorm

In order for Code Sniffer and Mess Detector to perform in PhpStorm they must first be integrated. This can be set as part of the default settings, meaning the settings will be applied to all projects, or in the project settings, where they would only valid for a specific project. The steps are exactly the same, but the entry point is slightly different. First access the settings using the menu item `File` and then selecting either `Default Settings...` or `Settings...` from the drop down menu.

PhpStorm Inclusion

To activate Code Sniffer click on `Languages & Frameworks`, `PHP` and then `Code Sniffer` or `Mess Detector`. In the text field next to "PHP Code Sniffer (phpcs) path" enter the path to phpcs.bat. In the standard installation this will be `C:\xampp\php\phpcs.bat`. After you have input the path, click on `Validate`. If the file is valid, the you will be shown a short text containing the version of the installed Code Sniffer. Although not specifically stated the steps for Mess Detector inclusion are completely analogous.

PhpStorm Configuration

Next we need to add PHP_CodeSniffer to the inspections performed. With the settings still open, click on `Editor` then on `Inspections`, then in the list to the right `PHP`, and finally on `PHP Code Sniffer validation` to open the configuration settings for Code Sniffer.

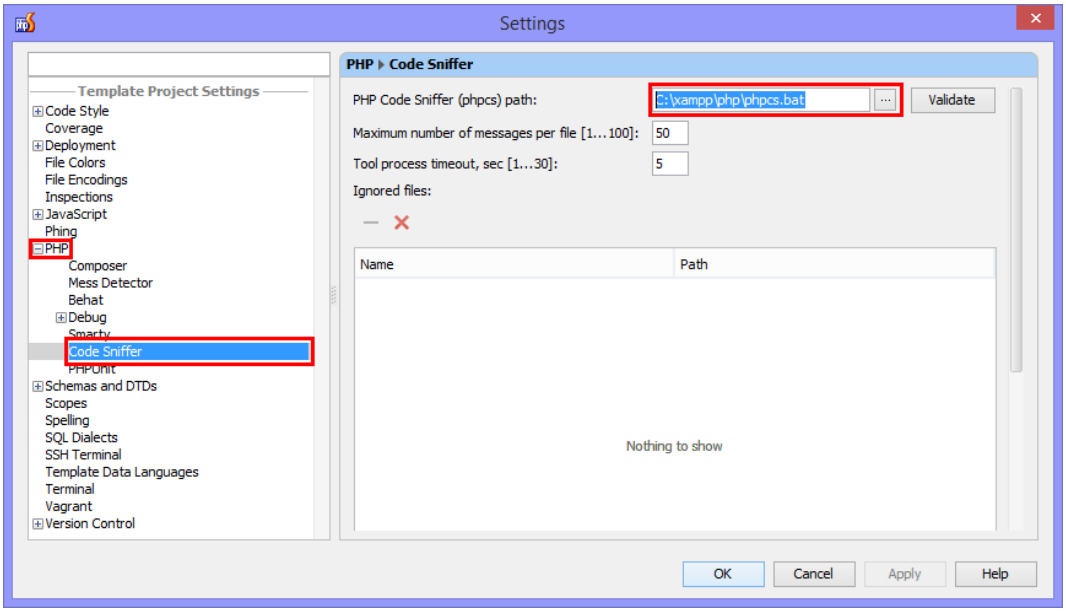

Figure 2.7: Code Sniffer - PhpStorm Inclusion

To activate the inspection we must first set the checkbox next to `PHP Code Sniffer validation`. This activation enables the further settings to the right which we see numbered with one through four in Figure 2.8.

1. `Inspection severity`:"indicates how seriously the code issues detected by the inspection impact the project and determines how the detected issues should be highlighted in the editor."[5]

2. `Display severity`: defines how the infractions found are marked.

3. `Refresh`: updates the list of available coding standards. Should the desired standard not be found in the folder defined in the section on symbolic links, you can also manually search for the standard on your file system.

4. `Available standards`: a list of standards found. In iCampus we use the THM Joomla Standard which inherits or extends many of the definitions in the Joomla standard.

It is recommended that both inspection and display severity be set to errors to make the standard infractions more noticable. Settings take effect when the `Apply` button has been pressed.

Inspection settings for Mess Detector are also analogous here, with the distinction that one need not select a standard, instead selecting which rule sets should be applied.

[5]http://www.jetbrains.com/phpstorm/webhelp/configuring-inspection-severities.html Stand September 22, 2014

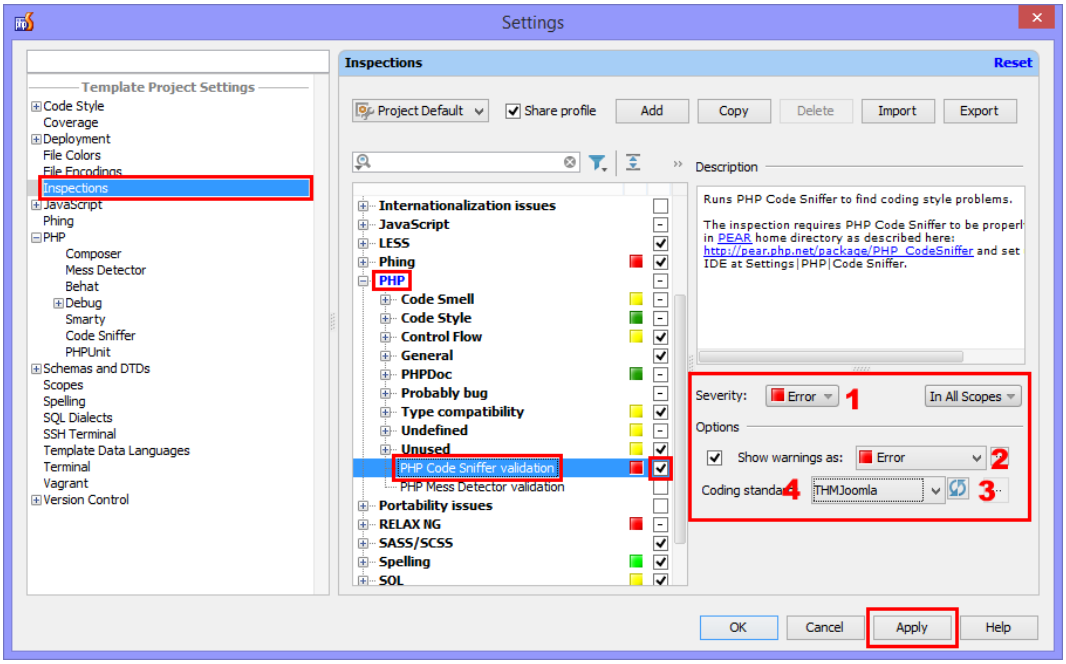

Figure 2.8: Code Sniffer - PhpStorm Configuration

2.5 Joomla!

Joomla is an award-winning content management system (CMS), which enables you to build Web sites and powerful online applications. Many aspects, including its ease-of-use and extensibility, have made Joomla the most popular Web site software available. Best of all, Joomla is an open source solution that is freely available to everyone.[6]

2.5.1 Joomla! 3.x Development

First extract the archive file downloaded from the Joomla! website and give it a URL friendly name such as "j3". Next place this folder in your web directory. If you followed the default installation steps this should be located at `C:\xampp\htdocs`.

Open a browser of your choosing and enter `localhost/<Your Joomla 3.x Folder>` in the address bar. If you renamed the backup "j3" this would then be `localhost/j3`. This will open the main configuration page as seen in Figure 2.9.

[6] `http://www.joomla.org/about-joomla.html` Stand 27 September, 2014

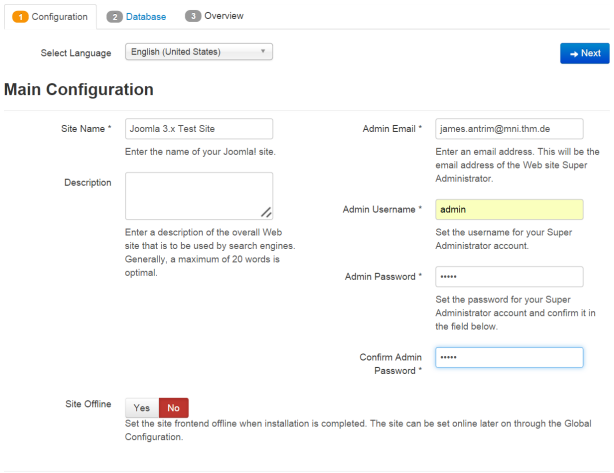

Figure 2.9: Joomla Main Configuration

Here you will need to enter the `Site Name`, `Admin Email`, `Admin Username`, `Admin Password`, and `Confirm Admin Password`. Click "⇒ Next" to move on to the database configuration.

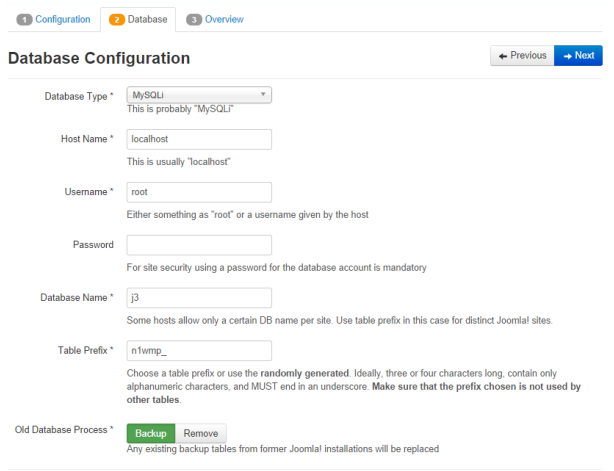

Figure 2.10: Joomla Database Configuration

Use the following values to complete the form, then click "⇒ Next" to continue on to the finalization.

`Database type`	MySQLi
`Host Name`	localhost
`Username`	root
`Password`	(empty, unless set)
`Database name`	j3 (unless named otherwise)

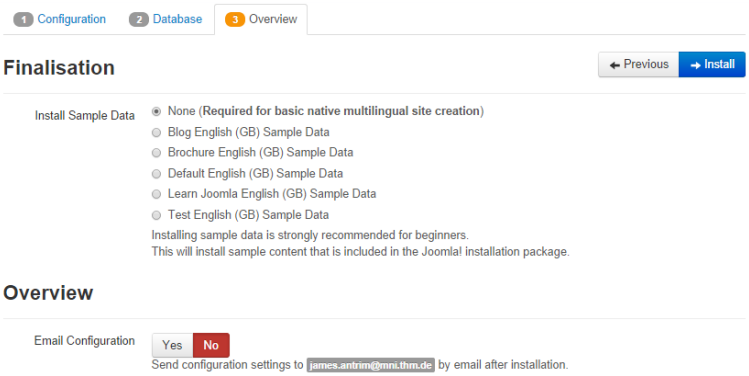

Figure 2.11: Joomla Finalization

Here you can choose whether or not you wish to install sample data. Leave "None" selected and click "⇒ Install" to complete the installation. The installation will then show the progress of the database construction. When this is finished you will be told that Joomla! is installed. You should now press the orange button "Remove installation folder" to complete the installation. You can choose to visit the frontend or backend of the site by clicking the appropriate button.