

SD203, Télécom ParisTech

PHP, MySQL, Framework, ORM: Lab assignment

Pierre Senellart (pierre.senellart@telecom-paristech.fr)

27 January 2016

The objective of this lab assignment is to discover Web development using PHP and MySQL with the help of a PHP framework (CodeIgniter) and an ORM (Doctrine). A simple news board will be developed to illustrate these technologies.

Handing out your assignment

The assignment must be handed out **by Wednesday, February 3rd, before 13:30:00, i.e., before the final lab session** (no extension will be given, later submissions will receive a 0). Put on the repository <http://services.infres.enst.fr/rendutp/depot/sd203-php/> an archive containing:

- all files (PHP, etc.) used to develop the site;
- a README.txt file with the URL of the home page of the created site and the assigned MySQL login.

Environment

You will put the Web site developed in the subdirectory `public.html` of your Unix home directory. If your Unix login is `pierre`, you can access a page `index.php` through a Web server at the URL: <http://www.infres.enst.fr/~pierre/index.php>.¹ Make sure the Web server has read access on your home directory and the sub-directory `public_html` with the command:

```
chmod a+rx ~ ~/public_html
```

You may need to reuse this command on newly created files, depending on your configuration.

We will use for this lab session the database management system MySQL/MariaDB available with the following parameters:

Server: `tiresias.enst.fr`

Login: of the form `tpn` (`tp1`, `tp2`, ..., `tp120`), as assigned during the lab session

Password: same as the login

Database: same as the login

These parameters can be used to connect to the database management system as follows:

¹You can also use URLs of the form <http://perso.telecom-paristech.fr/~pierre/index.php>. However, these will not work for the ORM Doctrine, as the PHP version installed on `perso.telecom-paristech.fr` is too old. For these reasons, it is recommended to use the `www.infres.enst.fr` Web server throughout the lab session.

using the **command line** with the command line tool `mysql` that can be used like this:

```
mysql -h server -u login -p database
```

with the **phpMyAdmin tool** at <http://tiresias.enst.fr/phpmyadmin/>

from the **CodeIgniter framework** by customizing the file `application/config/database.php`

One may (and should) refer throughout this lab assignment to the online documentation of:

PHP <http://php.net/>

MySQL <http://dev.mysql.com/doc>

CodeIgniter <https://codeigniter.com/userguide2/toc.html>

Doctrine <http://docs.doctrine-project.org/en/latest/>

1 CodeIgniter tutorial

In order to display PHP errors as they appear, create a `public_html/.htaccess` file in which you write the line:

```
php_flag display_errors on
```

1. Download Version 2 of the CodeIgniter PHP framework from <https://codeigniter.com/download>
2. Follow the instructions of <https://codeigniter.com/userguide2/installation/index.html> to install this framework in a subdirectory of your `public_html` directory (only the 4 numbered items, it is not necessary to follow the instructions regarding the `application` and `system` directories).
3. Carefully follow the tutorial (from introduction to conclusion) of the CodeIgniter framework to learn how to use this framework: <https://codeigniter.com/userguide2/tutorial/index.html>

Over the course of this tutorial you will develop a simple news board application, with many missing features. To avoid any caching issues, add the following line at the beginning of `index.php` (after the `<?php` pseudo-tag):

```
header('Expires: 0');
```

2 Making the news board usable

1. Customize the different views to make the whole application usable: there should be links from every page to the main page of the application, and it should be possible to access every page by a sequence of links from the home page of the application. The whole Web application should have a consistent appearance (but no need to add CSS markup).
2. Is the application vulnerable to HTML injections? Fix this.
3. Using the W3C validation service <http://validator.w3.org/>, make sure every page generated by the application is valid HTML code.
4. Add the possibility of deleting a news item by modifying the model and controller.
5. Right now an article is identified by its *slug* constructed from its title. What kind of problem can this scheme cause? Propose a solution and implement it.

3 Using the Doctrine ORM

1. Download the latest version of the DoctrineORM software from <http://www.doctrine-project.org/downloads/>
2. Carefully follow the instructions of <http://tinyurl.com/qgcubev> to set up Doctrine together with CodeIgniter, until the end of the section “Setup Entities, Proxies, and Database Schema” (don’t start the section “Using Doctrine in a CodeIgniter Controller”) taking into accounts the following points:
 - In the section “Add Models to Doctrine”, create a YAML description of the News item entity instead of the two entities of the tutorial (in the YAML markup, make sure to use a *table* name different from the table name already used in your CodeIgniter news board to avoid confusion); you can use a tool such as <http://www.yamllint.com/> to check the syntax of a YAML file.
 - In the section “Configure Doctrine Command Line Tool”, replace the line:

```
define('ENVIRONMENT', 'production');
```

with:

```
define('ENVIRONMENT', 'development');
```
 - In the section “Setup Entities, Proxies, and Database Schema”, use `~senellar/bin/php` instead of the `php` command line tool, which does not have all necessary libraries installed.
3. Modify your model to use Doctrine-managed objects instead of direct queries to the database. From CodeIgniter, you will be able to access the Doctrine entity manager with `$this->doctrine->em`. You will find a tutorial on how to store, retrieve, delete objects managed with Doctrine at <http://symfony.com/doc/current/book/doctrine.html>.
4. Modify your controllers and views accordingly. Test the Doctrine-powered Web site.

4 Completing the application

If you still have time left, add new features to the application. Some possible ideas:

- Add an authentication mechanism (login/password) to the application, with only authenticated users allowed to add news items.
- Retain in the database information about which user posted each news item, and make it so that a user can only delete her own news items.
- Add administrative roles to have a superuser able to manage all news items, manage users themselves, etc.

You can also come up with your own ideas of ways to extend the Web application.