



# Formation PHP - Symfony

Rush01

Rares Serban [rares.serban@pitechplus.com](mailto:rares.serban@pitechplus.com)  
42 pedago [pedago@42.fr](mailto:pedago@42.fr)

*Summary: You will create an intranet application similar to 42 Intranet.*

# Contents

<b>I</b>	<b>General Rules</b>	<b>2</b>
<b>II</b>	<b>Day-specific rules</b>	<b>3</b>
<b>III</b>	<b>Foreword</b>	<b>4</b>
<b>IV</b>	<b>Mandatory Part</b>	<b>5</b>
<b>V</b>	<b>Bonus Part</b>	<b>7</b>

# Chapter I

## General Rules

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- You must follow the **turn-in process** for each assignment. The url of your **GIT** repository for this day is available on your intranet.
- Your assignments will be evaluated by your Piscine peers.
- In addition to your peers evaluation, a program called the "Moulinette" should also evaluate your assignments. Fully automated, The Moulinette is tough and unforgiving in its evaluations. As a consequence, it is impossible to bargain your grade with it. Uphold the highest level of rigor to avoid unpleasant surprises.
- All shell assignments must run using `/bin/sh`.
- You must not leave in your turn-in repository any file other than the ones explicitly requested By the assignments.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the **mans** or on the Internet.
- Remember to use the Piscine forum of your intranet and also Slack!
- You must read the examples thoroughly. They can reveal requirements that are not obvious in the assignment's description.
- By Thor, by Odin! Use your brain!!!

# Chapter II

## Day-specific rules

- You **MUST** provide an author file named `auteur` at the root folder of your repository, as always.

```
$> cat -e auteur
noich$
thor$
$>
```

- This project will be entirely graded by humans. As such, you are free to have some variation between this subject's examples and your program's actual output. However, you do have to obey the spirit of the subject and turn in a program with consistent and relevant output formatting.
- For the same reasons, you are entirely free to choose your file names and general hierarchy. But keep in mind that a consistent and relevant file organization is a good code practice and will earn you bonus points.
- Try to respect the Symfony 2.8 coding standards as much as as possible, they can earn you bonus points.
- Other third party bundles or libraries can be used.
- Remember that no bonus points are awarded for the styling of the page, only the functionality and the code is all the matter. It you should at least try and organize the website and make everything easy to access.

# Chapter III

## Foreword

You will create an intranet for an organization of your choosing: company, school, government or others. The intranet will have multiple types of users and an administration backend to manage a variety of different aspects regarding it.

# Chapter IV

## Mandatory Part

Since this is an intranet, it will be only used internally by an organization. Because of this, no registration form will exist, but the users will be created by Administrators.

Your application will need to have two parts: the frontend, visible by your normal users, and a backend visible only to Administrators or other roles from where they can manage the website.

Set up security and an Administrator role that has access to the backend of the application. From the backend, he can create new users and assign them different roles. You can decide which roles can exist in your application. A user should have at least the following fields: first name, last name, email, password, created and role. The fields can be optional or not and they can have whatever types you consider fit.

You will also need to create a command which will allow you to create new Administrator users, since at the start your application will not have any users. After a user is created, he should receive a new confirmation email. From the email he can click a link and activate his account. On the activation page, he can enter a password for his account.

A logged in user has access to his profile page and also a form where he can edit his information. Make sure to also implement a forgot password functionality for the users. Other users can access the profile of the user and see his information. Make sure to properly implement security in your application and not let unauthorized users have access to specific actions.

Depending on what organization you chose, the user should also have access to a variety of sections. Some examples include: a list of all the groups/sections/courses he is part of, ability to register for a new group/section/course, a news section where latest news items can be displayed. Each section can have its own posts which can be created by Administrators or other roles from the backend area. Or instead of posts some other content can be used. It would be nice if you can also upload images when creating new content.

Regardless of what you choose, normal users should be able to comment on some content in your application and also reply to other user's comments. Also, you should implement pagination for content lists.

Users can also send private messages to other users from your application. These messages should not be real-time for now since that is included in the bonus part. You should also test the services you use in your application using unit testing.

You can use your imagination and expand the intranet how much you want. You will be rewarded depending on how much functionality you add.

# Chapter V

## Bonus Part

Implement ability for your users to post comments dynamically using Ajax calls. You can also implement Ajax calls in other parts of the application, for example let the users view/edit their profile without requiring a page reload.

Let users subscribe to sections/groups/categories and receive notifications when new content was added to that category. The notifications should be handled in real time using Websockets.

Implement a real time chat for your users using Websockets and also let users send photos and other files to each other. If sending a photo, a small preview should appear in the chat.