

1 Practice 9.2. Security

This practice is based on the previous one (9.1). Before start it, finish the previous one, make the corrections you consider, if needed, and start coding.

The mark of this practice **can increase the mark of the first evaluation by 1 point**, provided that the marks of the practice part and the exam are at least 5. The note that increases each exercise is indicated on it.

The maximum increase you can achieve this evaluation from voluntary practices is **1 point**, so if you do the practice 7.2 and this one, you can't get 1.5 points.

Before start, create a new git branch named `contacts-u9.2` and change to it:

```
git branch contacts-u9.2
git checkout contacts-u9.2
```

Do all the practice in the `contacts-u9.2` branch. Once finished, do the submission via GitHub Classroom as detailed in the end of the practice.

You can add documentation to explain what you have do, if needed. You can do a short document in PDF or a short (no more than 5 minutes) video showing how it works and explaining your job.

1.1 Exercise 1. OAuth

(+0,5 points)

Enable OAuth authentication using a provider of your choice (Google, Facebook, Twitter...). You can use the [HWIOAuthBundle](#)

1.2 Exercise 2. Token authentication

(+0,5 points)

Use token authentication to access the REST API. Do the login service via API, with the registered users you already have. Only the registered users could access to the API. If you want, you can implement the authorization system of the practice 9.1.

You can use the JWT token authentication system. some information:

- <https://www.sitepoint.com/php-authorization-jwt-json-web-tokens/>
- <https://www.techiediaries.com/php-jwt-authentication-tutorial/>

1.3 How to submit to GitHub Classroom

1. This task must be submitted to the same repository of the previous one. Remember to make a new branch before starting the practice.
2. Once you finish the task, make a commit with the comment "PRACTICE 9.2 SUBMISSION COMMIT", merge the branch into the main branch, and push it to GitHub. For example:

```
git commit -m "PRACTICE 9.2 SUBMISSION COMMIT"  
git checkout main  
git merge contacts-u9.2  
git push
```

3. Before that, you can do the commits and pushes you want. If you change your code after your submission commit, make another commit and push with the same text in the message adding the corrections you've done. Make a pull-request if you want so i could know your code is submitted.
4. Once pushed to GitHub, create a tag with the name "Practice 9.2". Create a new tag if you make corrections after the first submission (ex. "Practice 9.2 corrected").

If you have any doubt in your task, you can push your code and ask me by email what's your problem.