



ncsomori

(https://pr

(https://profile.intra.42.fr/)

(https://projects.intra.42.fr/scale\_teams/4217122/edit#)

(https://profile.intra.42.fr/)

My projects

(https://projects.intra.42.fr/)

projects.intra.42.fr/

Holy Graph

(https://projects.intra.42.fr/projects/g)

arning.intra.42.fr/)

List projects

(https://projects.intra.42.fr/projects/li

rflow.com/c/42network)

Available Coursus

(https://projects.intra.42.fr/cursus\_su

Your projects

panies.intra.42.fr/)

Day 02

(https://projects.intra.42.fr/projects/4:

meta.intra.42.fr/)

piscine-c-formation-piscine-php-

day-02)

hop.intra.42.fr/)

Day 04

(https://projects.intra.42.fr/projects/4:

piscine-c-formation-piscine-php-

day-04)

Day 05

(https://projects.intra.42.fr/projects/4:

piscine-c-formation-piscine-php-

day-05)

Day 06

(https://projects.intra.42.fr/projects/4:

piscine-c-formation-piscine-php-

day-06)

Day 07

(https://projects.intra.42.fr/projects/4:

piscine-c-formation-piscine-php-

day-07)

Rush00

(https://projects.intra.42.fr/projects/p

php-rush00)

ft\_printf

(https://projects.intra.42.fr/projects/ft

init

(https://projects.intra.42.fr/projects/in

Piscine PHP

(https://projects.intra.42.fr/projects/p

php)

## SCALE FOR PROJECT PISCINI (HTTPS://PROJECTS.INTRA.42.FR/PI PHP) / DAY 04

## (HTTPS://PROJECTS.INTRA.42.FR/P PISCINE-C-FORMATION-PISCINE-PI

You should evaluate 1 student in this team



Git repository

git@vogosphere-v2.hive.fi:vogsp

## Guidelines

You already know how peer2peer evaluation work by now and during this PHP Piscine, it works as any other Piscine projects you have evaluate so far. Take the time to discuss the assignments and to make sure that the evaluated has understood the notions of the day.

## Attachments

### User sessions (EN subs)

User sessions (EN subs) (https://cdn.intra.42.fr/video/video/1215/pisc SUBS.mp4)

☐ Subject (https://cdn.intra.42.fr/pdf/pdf/10079/d04.er

## Preliminaries

There are a lot of exercises, you will have to manage your time according to the fastest and most efficient tests. Be careful, this evaluation form won't be a test, you have to come up with your own! As you most likely have created your own assignments, feel free to re-use them. Once an exercise is wrong but you can still discuss the rest of the assignments.

### The basics

- The whole team must be present.
- Only grade the work that is in the student or group's Git repository.
- You must use Chrome for this defense.
- If there is nothing in the repository, the evaluation stops here. Discuss what went wrong and how to avoid facing the same situation tomorrow.

☐ Yes

## Ex00

## Session

Execute the examples provided in the subject and then make your own tests.

Check that `index.php` contains a form which uses the GET method, the `login`, the password field `passwd`, the validation button is called `submit`. If that part is correct, add **3 points**.

Example of output:

```
$> curl -v -c cook.txt 'http://localhost:8080/d04/ex00/in
...
> GET /ex00/index.php HTTP/1.1
[...]
> Host: localhost:8080
> Accept: */*
>
< HTTP/1.1 200 OK
[...]
* Added cookie PHPSESSID="ji22ln1iua24au9eo97ubbego0" for
path /, expire 0
< Set-Cookie: PHPSESSID=ji22ln1iua24au9eo97ubbego0; path=
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
[...]
<html><body>
<form method="get" action="index.php">
  Username: <input type="text" name="login" value="" />
  <br />
  Password: <input type="password" name="passwd" value=""
  <input type="submit" name="submit" value="OK" />
</form>
</body></html>
$>
```

Check that the submitted `login` and `passwd` values are saved in the session values in the previous form. If that part is correct, add **2 points**.

Example of output:

```
$> curl -v -b cook.txt 'http://localhost:8080/d04/ex00/index.php?login=sb&passwd=beeone&submit=OK'
[...]
```

```
>
GET /ex00/index.php?login=sb&passwd=beeone&submit=OK HTTP
[...]
```

```
>
Host: localhost:8080
> Accept: */*
> Cookie: PHPSESSID=ji22ln1iua24au9eo97ubbego0
>
< HTTP/1.1 200 OK
[...]
```

```
<
<html><body>
<form method="get" action="index.php">
  Username: <input type="text" name="login" value="sb" />
  <br />
  Password: <input type="password" name="passwd" value="" />
  <input type="submit" name="submit" value="OK" />
</form>
</body></html>
$>
```

Rate it from 0 (failed) through 5 (excellent)

5

## Ex01

### Create account

Execute the examples provided in the subject and then make your own tests.

- Check that a simple account creation works, try to create several new accounts must be stored in /private/passwd  
If that part is correct, add **2 points**.

```
$> curl -d login=truc -d passwd=titi1 -d submit=OK 'http://localhost:8080/d04/ex01/create.php'
OK
$> more /path/to/mamp/apache2/htdocs/d04/private/passwd
a:1:{i:0;a:2:{s:5:"login";s:4:"truc";s:6:"passwd";s:128:"37ac1b4ca7908a431019e8b93c9fd337317f92fac80dace29802bedc3a6c1df8579cdaa02256099ed52a905d38";}}
```

Check the error management: it should not be possible to use the same username and a blank password.

If that part is correct, add **2 points**.

```
$> curl -d login=truc -d passwd=titi1 -d submit=OK 'http:
4/ex01/create.php'
ERROR
$> curl -d login=machin -d passwd= -d submit=OK 'http://l
x01/create.php'
ERROR
$>
```

- If the password is stored in plain text, the exercise is KO and the evaluac
- Check that the hash algorithm used for the password is not md5 or sha' add **1 point**.

Rate it from 0 (failed) through 5 (excellent)

5

## Ex02

### Modif account

Execute the examples provided in the subject and then make your own te

- Remove the `/private/passwd` file to prevent any problem.
- Create an account using the `create.php` file that must be present in the

```
$> rm /path/to/mamp/apache2/htdocs/d04/private/passwd
$> curl -d login=z -d passwd=21 -d submit=OK 'http://loca
/create.php'
OK
```

- Change the password and check that it has been updated in `/private/`. If that part is correct, add **2 points**.

```
$> curl -d login=z -d oldpw=21 -d newpw=42 -d submit=OK '
80/d04/ex02/modif.php'
OK
$> more /path/to/mamp/apache2/htdocs/d04/private/passwd
a:1:{i:0;a:2:{s:5:"login";s:1:"z";s:6:"passwd";s:128:"769
036bce3a29e7667d5a228a85ea825210d3944c2fad2f15c36aa6affa7
a199f0a9f45780c64261f1b22c0b7f";}}
```

- Check that an error is returned when providing a wrong password. Also be possible to put an empty password. If that part is correct, add **2 points**.

```
$> curl -d login=x -d oldpw=12 -d newpw=21 -d submit=OK '80/d04/ex02/modif.php'
ERROR
$> curl -d login=x -d oldpw=42 -d newpw= -d submit=OK 'http://localhost:8080/d04/ex02/modif.php'
ERROR
```

- If the password is stored in plain text, the exercise is KO and the evaluator will not add 1 point.
- Check that the hash algorithm used for the password is not md5 or sha1. If that part is correct, add 1 point.

Rate it from 0 (failed) through 5 (excellent)

5

## Ex03

### Auth

Execute the examples provided in the subject and then make your own tests.

- Create a new account and test the login.php : a wrong login/password fail, a good one should succeed and set the logged\_on\_user session variable with the file whoami.php.

If that part is correct, add 3 points.

```
$> rm /path/to/mamp/apache2/htdocs/d04/private/passwd
$> curl -d login=toto -d passwd=titi -d submit=OK 'http://localhost:8080/ex01/create.php'
OK
$> curl -c cook.txt 'http://localhost:8080/d04/ex03/login.php?passwd=titi'
OK
$> curl -b cook.txt 'http://localhost:8080/d04/ex03/whoami.php'
toto
$>
```

- The login.php file must include auth.php and use the auth function. If that part is correct, add 1 point.
- The logout.php must properly destroy the session cookie. If that part is correct, add 1 point.

```
$> curl -b cook.txt 'http://localhost:8080/d04/ex03/logout.php'
$> curl -b cook.txt 'http://localhost:8080/d04/ex03/whoami.php'
ERROR
$>
```

Rate it from 0 (failed) through 5 (excellent)



## Ex04

### 42chat - Part 1/2

Execute the examples provided in the subject and then make your own tests.

- Check that the `index.html` page has a sign-in form, there is a link to a link to modify an existing account, if so, add **1 point**.
- Check that once registered, it's possible to sign in and after doing so, there are iframes, if so, add **1 point**.

```
$> curl -d login=user1 -d passwd=pass1 -d submit=OK 'http://localhost:8080/d04/ex04/create.php'
OK
$> curl -d login=user2 -d passwd=pass2 -d submit=OK 'http://localhost:8080/d04/ex04/create.php'
OK
$> curl -c user1.txt -d login=user1 -d passwd=pass1 -d submit=OK 'http://localhost:8080/d04/ex04/login.php'
[...]
```

- Check that the bottom iframe contains a form that uses `speaks.php` to add a message in the chat. If that part is correct, add **1 point**.

```
$> curl -b user1.txt -d submit=OK -d msg=Bonjours 'http://localhost:8080/d04/ex04/speak.php'
[...]
```

- Check that the `/private/chat` file is filled with every new message, if so, add **1 point**.

```
$> more /path/to/mamp/apache2/htdocs/d04/private/chat
a:2:{i:0;a:3:{s:5:"login";s:5:"user1";s:4:"time";i:136428;"Bonjours";}}i:1;a:3:{s:5:"login";s:5:"user2";s:4:"time";i:136429;"Hello";}}
```

- Check that the `chat.php` file reads and displays the content of the `/private/chat` file in proper format. If that part is correct, add **1 point**.

```
$> curl -b user2.txt 'http://localhost:8080/d04/ex04/chat'
[09:42] <b>user1</b>: Bonjours<br />
[09:43] <b>user2</b>: Hello<br />
$>
```



Rate it from 0 (failed) through 5 (excellent)

## 42chat - Part 2/2

- Check that after submission of the `speak.php` form, the other iframe (c add **1 point**.
- Check that it's possible to log out, after which it's not possible to post ar `speak.php` . If that part is correct, add **1 point**.

```
$> curl -b user1.txt -c user1.txt 'http://localhost:8080/
'
$> curl -b user1.txt -d submit=OK -d msg=Bonjours 'http:/
/ex04/speak.php'
ERROR
$>
```

- Check that **flock** was used in the `speak.php` file to prevent two differe simultaneously writing a message in the `/private/chat` file. If that part
- Check that **flock** was used in the `chat.php` file to prevent reading the it's being modified. If that part is correct, add **1 point**.



Rate it from 0 (failed) through 5 (excellent)

## Ratings

Don't forget to check the flag corresponding to the defense

<input type="checkbox"/> Ok			
<input type="checkbox"/> Empty work	<input type="checkbox"/> Cheat	<input type="checkbox"/> Crash	<input type="checkbox"/> 1

## Conclusion

Leave a comment on this evaluation

Finish evaluation

Terms of use for video surveillance  
(<https://signin.intra.42.fr/legal/terms/1>)

Rules of procedure  
(<https://signin.intra.42.fr/legal/terms/4>)

Declaration on the use of cookies  
(<https://signin.intra.42.fr/legal/terms/2>)

General term of us  
(<https://signin.intra.42.f>)