# Simon Gomez - Microservices

## Authentication app

Goal of this TD is to implement a simple authentication in a new nodejs App

Target is to have :

- a login form

| | |
|---|---|
| **Login** | *simon.gomez* |
| **Password** | **** |

- a register form

| | |
|---|---|
| **Login** | *simon.gomez* |
| **Password** | **** |
| **Password** | **** |

## Step 1 : session management

set up a session management using the express-session middleware

- add the express-session to your application
- findout how the session is kept between request
- create an api /session that will display the content of the session

TIPS :

- you can use JSON.stringify to display information
- session is available on the server side : req.session

it could be a good idea to follow the express good practice regarding the session : [http://expressjs.com/en/advanced/best-practice-security.html#dont-use-the-default-session-cookie-name](http://expressjs.com/en/advanced/best-practice-security.html#dont-use-the-default-session-cookie-name)

## Step 2 : login form

make sure your app is secure :

- if a user is not logged in, redirect him to the login page
- it should input a login and a password
- the login/password should be verified using a variable or a flat file or a database (depending on your progress)
    - if not correct : display an error message
    - if correct, save the login in the session and display the game main page

TIPS :

- you can use the following code to controle the app login :

```
- app.use(req,res,next)=>{
    if(req.session.user){
      next()
    }else{
      res.redirect("/login.html)
    }
 })
```

## Step 3 : register form

allow your user to register create a register form that request a login and a password

- if login does not exist in your referentiel, add the login/password to your storage (variable / flat file/ database)
- if the login already exist, display an error message

## Test Everything together

- try connect with a non registered user
- ensure it fail
- register the user
- try connect with a bad password
- ensure it fail
- try connect with a valid password
- ensure it work

## Security consideration

if your password is stored in plain text, update your code to use a hash to store it. you can use sha256 with the crypto lib to store the password.

For webaps, lots of risk are to be covered. A good starting point : https://blog.risingstack.com/node-js-security-checklist/

Published with GitHub Pages