

# Spring Boot :

## Atelier OAuth2 : Social Login


### “Connect With GITHUB”


#### Objectifs :


1. Inscription de l'application sur le serveur d'autorisation de GITHUB
2. Création d'un projet Spring boot,
3. Tester la sécurisation d'une api à travers de login GITHUB

#### Inscription de l'application sur le serveur d'autorisation de GITHUB

Connectez-vous à GITHUB, puis settings/Developer Settings

 GitHub Apps

 OAuth Apps

 Personal access tokens



New OAuth App

Cliquez sur

## Register a new OAuth application

Application name \*

hello-app

Something users will recognize and trust.

Homepage URL \*

http://localhost:8080

The full URL to your application homepage.

Application description

Tester hello api

This is displayed to all users of your application.

Authorization callback URL \*

http://localhost:8080/api/hello

Your application's callback URL. Read our [OAuth documentation](#) for more information.

☐ Enable Device Flow

Allow this OAuth App to authorize users via the Device Flow.

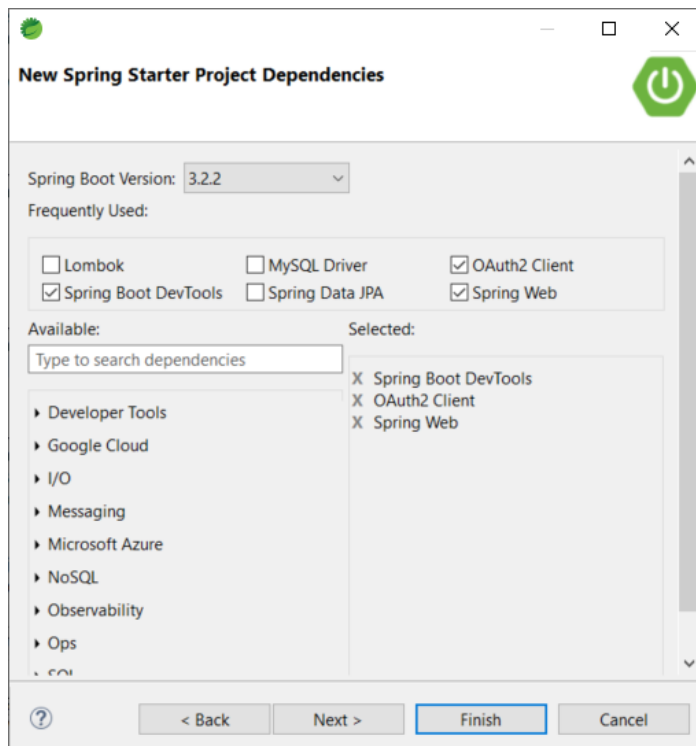
Read the [Device Flow documentation](#) for more information.

Register application

Cancel

Copier votre CLIENT\_ID et votre CLIENT\_SECRET dans un fichier texte, afin de les utiliser plus tard dans le fichier application.properties de votre application spring boot

## Créer le projet spring boot



## 1. Créer la classe MyController contenant l'api

```
package com.nadhem.oauth2.web;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api")
public class MyController {

    @GetMapping("/hello")
    public ResponseEntity<String> hello(){

        return ResponseEntity.ok("Salut tu es bien authentifié !");
    }
}
```

## 2. Tester l'api /api/hello

http://localhost:8080/api/hello

## 3. Créer la classe SecurityConfig

```
package com.nadhem.oauth2.security;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.Customizer;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain (HttpSecurity http) throws
    Exception
    {
        http.authorizeHttpRequests((requests)-requests
            .anyRequest().authenticated())
            .oauth2Login(Customizer.withDefaults());
        return http.build();
    }
}
```

## 4. Ajouter ces lignes au fichier application.properties

```
spring.security.oauth2.client.registration.github.client-id=*****  
spring.security.oauth2.client.registration.github.client-secret=*****
```

A remplacer les \* par vos valeurs obtenues lors de l'inscription dans GITHUB  
Authorization Server

#### 5. Tester l'api /api/hello

<http://localhost:8080/api/hello>