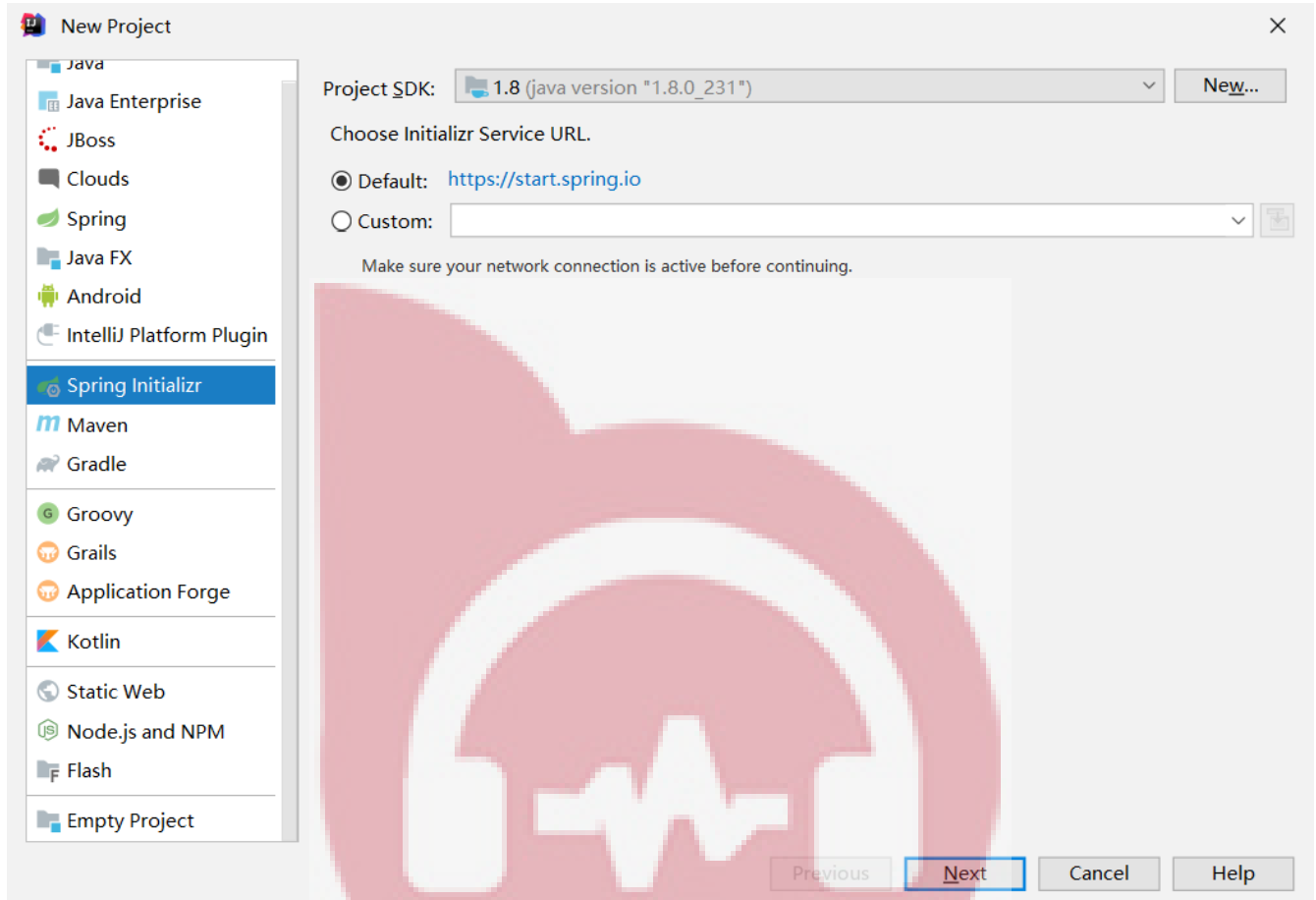


## Spring Security Oauth2 整合单点登录 (SSO)

### 创建客户端





New Project

Project Metadata

Group:

com.xxxx

Artifact:

oauth2client01demo

Type:

Maven Project (Generate a Maven based project archive.)

Language:

Java

Packaging:

Jar

Java Version:

8

Version:

0.0.1-SNAPSHOT

Name:

oauth2client01demo

Description:

Demo project for Spring Boot

Package:

com.xxxx.oauth2client01demo

Previous

Next

Cancel

Help

New Project

Project name:

oauth2client01demo

Project location:

D:\workspace\demo\oauth2client01demo

...

More Settings

Previous

Finish

Cancel

Help

## 添加依赖



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.2.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.xxxx</groupId>
  <artifactId>oauth2client01demo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>oauth2client01demo</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
    <spring-cloud.version>Greenwich.SR2</spring-cloud.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-oauth2</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-security</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>io.jsonwebtoken</groupId>
      <artifactId>jjwt</artifactId>
      <version>0.9.0</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
```



```
<artifactId>spring-cloud-dependencies</artifactId>
<version>${spring-cloud.version}</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

## 修改配置文件

application.properties

```
server.port=8081
#防止Cookie冲突，冲突会导致登录验证不通过
server.servlet.session.cookie.name=OAUTH2-CLIENT-SESSIONID01
#授权服务器地址
oauth2-server-url: http://localhost:8080
#与授权服务器对应的配置
security.oauth2.client.client-id=admin
security.oauth2.client.client-secret=112233
security.oauth2.client.user-authorization-uri=${oauth2-server-url}/oauth/authorize
security.oauth2.client.access-token-uri=${oauth2-server-url}/oauth/token
security.oauth2.resource.jwt.key-uri=${oauth2-server-url}/oauth/token_key
```

## 在启动类上添加@EnableOAuth2Sso注解来启用单点登录功能

```
package com.xxxx.oauth2client01demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.security.oauth2.client.EnableOAuth2Sso;

@SpringBootApplication
@EnableOAuth2Sso
public class Oauth2client01demoApplication {

    public static void main(String[] args) {
        SpringApplication.run(Oauth2client01demoApplication.class, args);
    }

}
```



## 添加接口用于获取当前登录用户信息

```
package com.xxxx.oauth2client01demo.controller;

import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/user")
public class UserController {

    @GetMapping("/getCurrentUser")
    public Object getCurrentUser(Authentication authentication) {
        return authentication;
    }

}
```

## 修改认证服务器配置

修改授权服务器中的AuthorizationServerConfig类，将绑定的跳转路径为

<http://localhost:8081/login>，并添加获取秘钥时的身份认证

```
@Override
public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
    clients.inMemory()
        //配置client_id
        .withClient("admin")
        //配置client-secret
        .secret(passwordEncoder.encode("112233"))
        //配置访问token的有效期
        .accessTokenValiditySeconds(3600)
        //配置刷新token的有效期
        .refreshTokenValiditySeconds(864000)
        //配置redirect_uri，用于授权成功后跳转
        // .redirectUri("http://www.baidu.com")
        //单点登录时配置
        .redirectUri("http://localhost:8081/login")
        //配置申请的权限范围
        .scopes("all")
        //自动授权配置
        .autoApprove(true)
        //配置grant_type，表示授权类型
        .authorizedGrantTypes("authorization_code", "password", "refresh_token");
}

@Override
public void configure(AuthorizationServerSecurityConfigurer security) {
    // 获取密钥需要身份认证，使用单点登录时必须配置
}
```



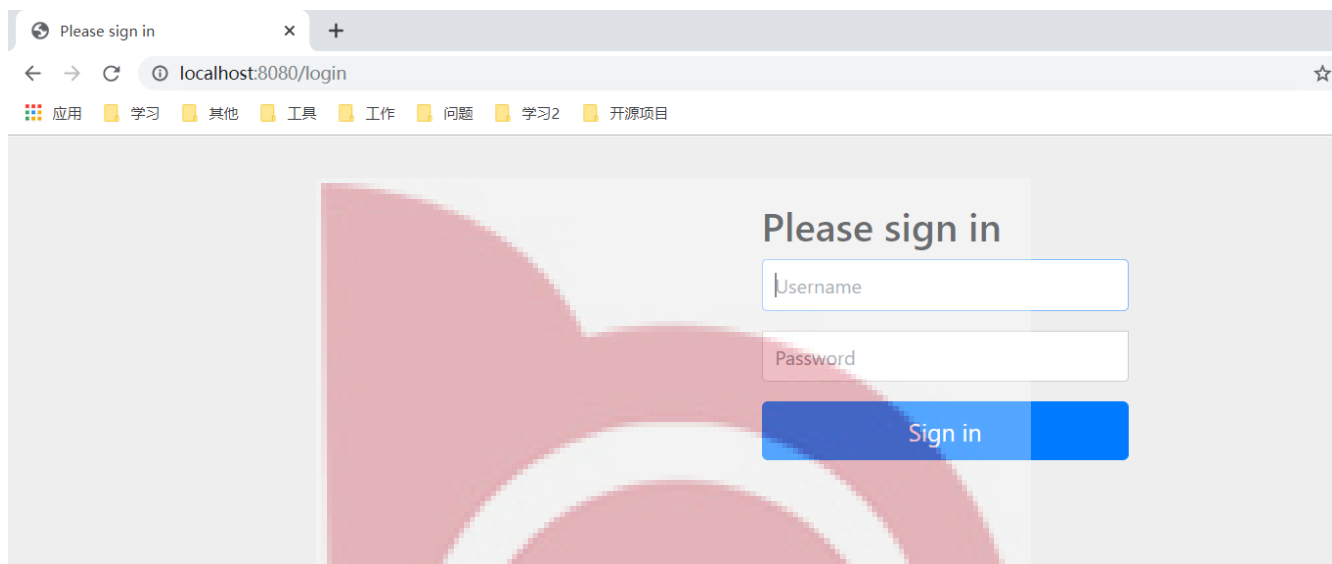
```
security.tokenKeyAccess("isAuthenticated()");  
}
```

## 7、测试

启动授权服务和客户端服务；

访问客户端需要授权的接口<http://localhost:8081/user/getCurrentUser>

会跳转到授权服务的登录界面；



授权后会跳转到原来需要权限的接口地址，展示登录用户信息；

