

# 在Spring boot应用HiaCloud UAA

以极简的方式，在Spring boot项目中使用UAA进行身份验证。

## 获取UAA Client信息

如果您还没有UAA Client和token key，可通过以下方式创建：

### 1. 创建UAA Client

```
POST http://192.168.66.158:18080/oauth/client/add
{
  "authorities": "openid",
  "authorizedGrantTypes": "authorization_code,refresh_token,client_credentials,password",
  "autoapprove": "false",
  "clientId": "MyClient",
  "clientSecret": "MyClient",
  "scope": "openid",
  "webServerRedirectUri": "http://localhost:8080/login"
}
```

其中：192.168.66.158:18080是UAA地址及端口。

### 1. 创建Token Key

```
GET http://192.168.66.158:18080/oauth/token_key
{
  "alg": "SHA256withRSA",
  "value": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnGp/Q5lh0P8nPL21oMMrt2RrkT9AW5jgYwLfSunJvc9G6uR3cXRRDCjHqWU5WYwivcF180A
6CWp/ireQFFBNowgc5XA0kPpzEtgsA5YsNX7iSnUibB004iBTfU9hZ2Rbsc8cWqynT0RyN4TP1RYVSeVKvMQk4GT1r7JCEC+Tnu1ELmbNwMQyzKjsfBX
yIOCFU/E94ktvsTZUHF40q44DBylCDsS1k7/sfZC2G5EU70z0mhG8+Uz6MSEQHtoIi6mc8u64Rwi3Z3tscuWG2ShtsUFuNSAFNkY7LkLn+/hxLCu2bNIS
MaESa8dG22CIMuIeRLVcAmEWEWH5EEforTg+QIDAQAB\n-----END PUBLIC KEY-----"
}
```

value部分即为token key

## 创建Spring boot web项目

略

## Oauth2 安全配置

### 1. 配置pom.xml

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.security.oauth</groupId>
<artifactId>spring-security-oauth2</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.security</groupId>
```

```
<artifactId>spring-security-jwt</artifactId>
</dependency>
```

## 2. 在Java中配置Security

```
import org.springframework.boot.autoconfigure.security.oauth2.client.EnableOAuth2Sso;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.web.csrf.CookieCsrfTokenRepository;

@Configuration
@EnableOAuth2Sso
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        // @formatter:off
        http.authorizeRequests()
            .antMatchers("/", "/login**", "/webjars/**").permitAll()
            .anyRequest().authenticated()
            .and().logout().logoutSuccessUrl("/").permitAll()
            .and().csrf().csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse());
        // @formatter:on
    }
}
```

## 3. 配置application.yml

```
security:
  oauth2:
    client:
      accessTokenUri: http://192.168.66.158:18080/oauth/token
      userAuthorizationUri: http://192.168.66.158:18080/oauth/authorize
      clientId: wanghq
      clientSecret: wanghq
    resource:
      jwt:
        keyValue: |
          -----BEGIN PUBLIC KEY-----
          MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnGp/Q5lh0P8nPL2loMMrt2RrkT9AW5jgYwLfSUnJvc9G6uR3cXRRDCjHqWU5WYwivcF180A6CWp/ir
          eQFFBNowgc5XaA0kPpzEtgsA5YsNX7iSnUibB004iBTfU9hZ2RbSc8cWqynT0RyN4TP1RYVSeVKvMQk4GT1r7JCEC+Tnu1ELmbNwMQyzKjsfBXyIOCFU/E94kt
          vsTZUHF40q44DBylCDsS1k7/sfZC2G5EU70z0mhG8+Uz6MSEQHtoIi6mc8u64Rwi3Z3tsCuWG2ShTsUFuNSAFNkY7LkLn+/hxLCu2bNISMaESa8dG22CIMuIeR
          LVcAmEWEWH5EEforTg+QIDAQAB
          -----END PUBLIC KEY-----
```

## 参考代码

<https://github.com/hiacloud/examples/tree/master/spring-boot-uaa>