# 一、SpringSecurity做密码的生成和校验

## 1.引入依赖

<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-security</**artifactId**>  
</**dependency**>

## 2.配置bean

|  |
| --- |
| @Configuration  @EnableWebSecurity  public class WebSecurityConfig extends WebSecurityConfigurerAdapter {  @Override  protected void configure(HttpSecurity http) throws Exception {  http.authorizeRequests()  .antMatchers("/\*\*").permitAll()  .anyRequest().authenticated()  .and().csrf().disable();  }  @Bean  public BCryptPasswordEncoder bcryptPasswordEncoder(){  return new BCryptPasswordEncoder();  }  } |

## 3.调用方法

1）生成MD5密文：

encode()

2)匹配密文：

mathes()

# 二、单点登录系统

## 1.登录验证成功后往redis中存入对象

此时单点登录系统中的redis就好比单体应用中的session

因此需要把登录验证后的user对象存入到redis中，键： user:token:uuid

## 2.把uuid存入到客户端cookie中

当客户端访问其他服务器时能携带uuid.

|  |
| --- |
| @PostMapping(**"checkLogin"**) @ResponseBody **public** String checkLogin(TUser tUser, HttpServletResponse response){  *//找servier* TUser currentUser = **userService**.checkLogin(tUser);  **if**(currentUser!=**null**){  *//===登录成功后我们要做什么事情?==  //1.往redis里存对象* String uuid = UUID.*randomUUID*().toString();  *//设置key的序列化器，不是必须的* **redisTemplate**.setKeySerializer(**new** StringRedisSerializer());   String redisUserTokenKey = **new** StringBuilder().append(RedisConstant.***userToken***).append(uuid).toString();    *//事不过三 三则重构* **redisTemplate**.opsForValue().set(redisUserTokenKey,currentUser);  *//设置一个超时时间* **redisTemplate**.expire(redisUserTokenKey,30, TimeUnit.***MINUTES***);   *//2.往客户端的cookie中存放uuid* Cookie cookie = **new** Cookie(CookieConstant.***userToken***,uuid);  *//cookie可以操作的： path maxage* response.addCookie(cookie);     **return "ok"**;  }  **return "not ok"**; } |

## 3.验证是否已登录

思路：

1）先从客户端获取键是“user\_token”的那个cookie

2）如果有的话得到其值 uuid

3）根据uuid去redis里拿 user对象

4）如果拿到user对象，已登录

5）否则，未登录

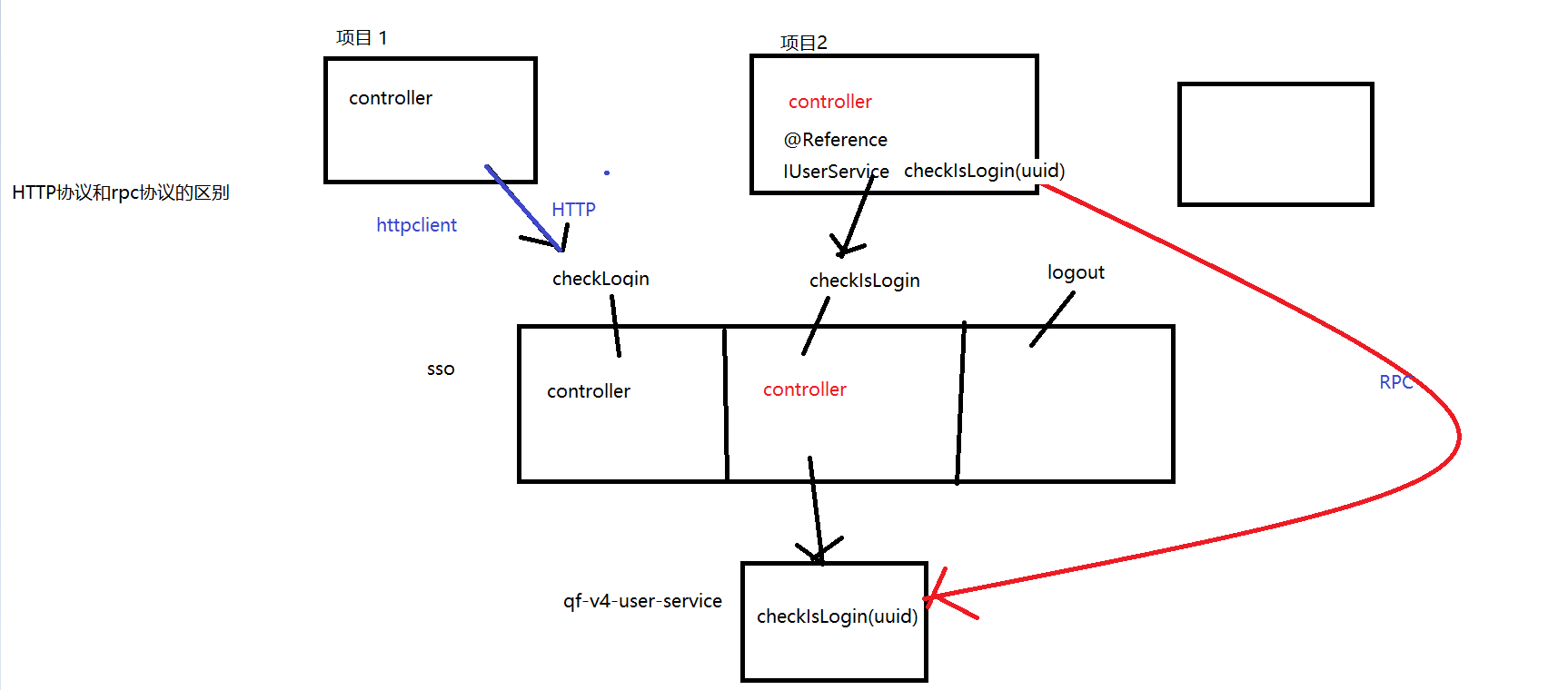
## 4.注销

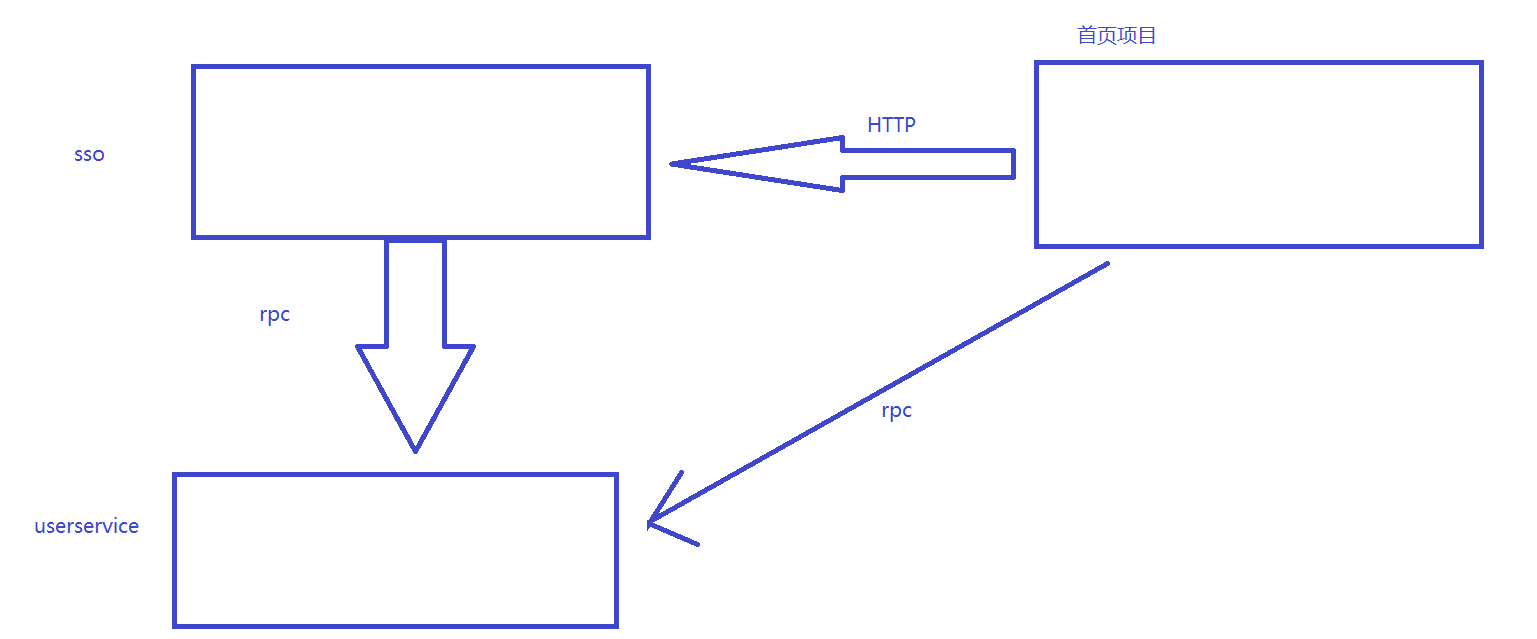
思路：

1）根据uuid删除redis中的键值对

2）删除cookie中的键值对

# 三、http协议和rpc协议的区别





# 四、首页的无级分类

参考文档

# 五、springboot如何解决跨域

参考文档