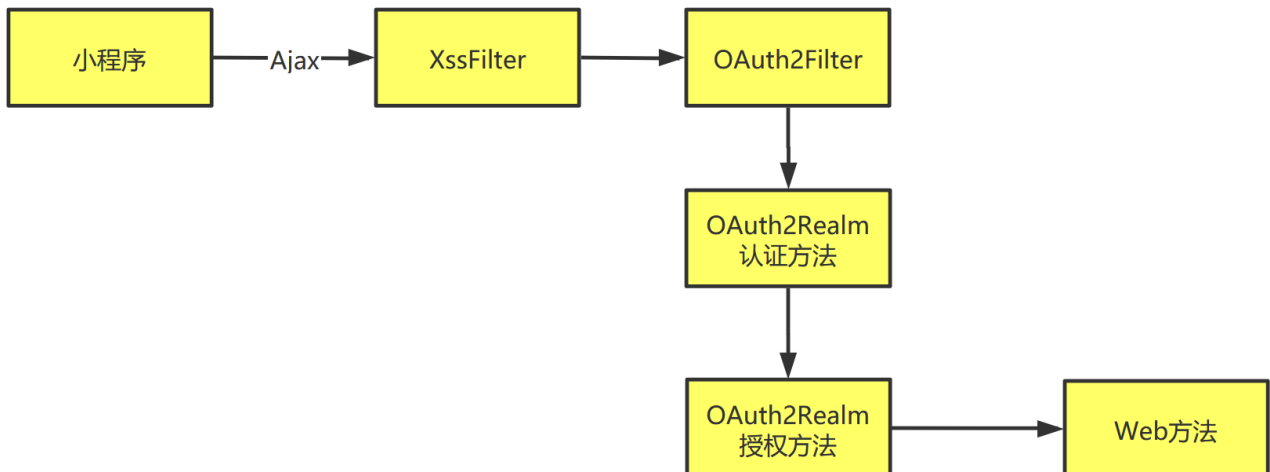


4-10 实现Shiro认证功能

140.143.132.225:8000/project-1/doc-58

一、认证与授权执行流程

之前我们在配置Shiro+JWT的时候，Shiro的认证与授权的实现功能并没有完成。



现在Controller中的 `validCanCheckIn()` 方法，并不是Shiro放行请求的Web方法。所以发送给 `validCanCheckIn()` 方法的请求一定会被Shiro拦截下来，先由 `OAuth2Filter` 检查请求头的Token是否合法。如果没问题，接下来就要由 `OAuth2Realm` 中的 `doGetAuthenticationInfo()` 方法来颁发认证对象。请求被赋予了认证对象，那么请求才会被发送到Web方法来执行。

二、查询用户信息

因为在认证方法里面要返回认证对象，认证对象创建的时候要传入用户信息和令牌，所以我们这里就要查询用户信息，然后判断用现在是在职还是离职状态。如果是在职状态，那就可以创建认证对象，反之就抛出异常。

编辑 `TbUserDao.xml` 文件，添加查询方法

```
1. <select id="searchById" parameterType="int"
   resultType="com.example.emos.wx.db.pojo.TbUser">
2.     SELECT
3.         id, open_id, nickname, photo, name, sex, tel, role, root, dept_id, status,
   create_time
4.     FROM tb_user WHERE id=#{userId} AND status = 1
5. </select>
```

编辑 `TbUserDao.java` 接口，添加抽象方法

```
1. public TbUser searchById(int userId);
```

编辑 `UserService.java` 接口，添加抽象方法

```
1. public TbUser searchById(int userId);
```

编辑 `UserServiceImpl.java` 类，实现抽象方法

```
1. @Override
2. public TbUser searchById(int userId) {
3.     TbUser user = userDao.searchById(userId);
4.     return user;
5. }
```

二、实现认证方法

编辑 `OAuth2Realm.java` 文件，修改 `doGetAuthenticationInfo()` 方法

```
1. /**
2.  * 认证(登录时调用)
3.  */
4. @Override
5. protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken token) throws
    AuthenticationException {
6.     String accessToken = (String) token.getPrincipal();
7.     int userId = jwtUtil.getUserId(accessToken);
8.     //查询用户信息
9.     TbUser user = userService.searchById(userId);
10.    if(user==null){
11.        throw new LockedAccountException("账号已被锁定,请联系管理员");
12.    }
13.    SimpleAuthenticationInfo info=new SimpleAuthenticationInfo(user, accessToken,
        getName());
14.    return info;
15. }
```

