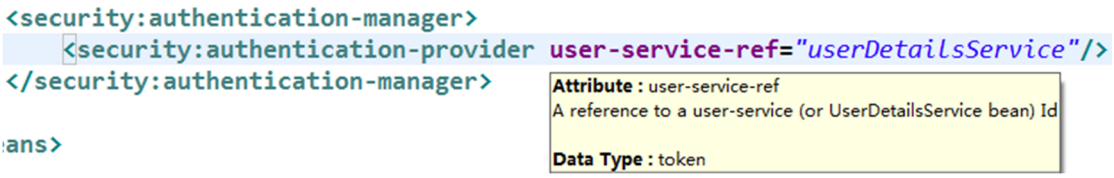
SpringSecurity 之自定义用户权限信息的存取

1. SpringSecurity 版本为：3.1.0
2. 在使用配置文件的情况下，使用 security:user-service 节点配置用户信息：用户民，密码，用户所有的权限

```
<!-- 配置用户信息:用户名, 密码以及该用户所有的权限 -->
<security:user-service id="userService">
    <security:user name="admin" password="admin" authorities="ROLE_ADMIN,ROLE_USER"/>
    <security:user name="user" password="user" authorities="ROLE_USER"/>
</security:user-service>

<!-- 配置 SpringSecurity 时, 必须配置 authentication-manager: 在该配置项中配置具体的用户信息 -->
<security:authentication-manager>
    <!-- authentication-provider: 通过 user-service-ref 指向实际的用户信息 -->
    <security:authentication-provider user-service-ref="userService"/>
</security:authentication-manager>
```

3. security:authentication-provider 节点的 user-service-ref 属性的提示为：



提示信息意为：可以将 user-service-ref 属性值指向一个 UserDetailsService 的 Bean。在其中实现自定义用户信息的存取

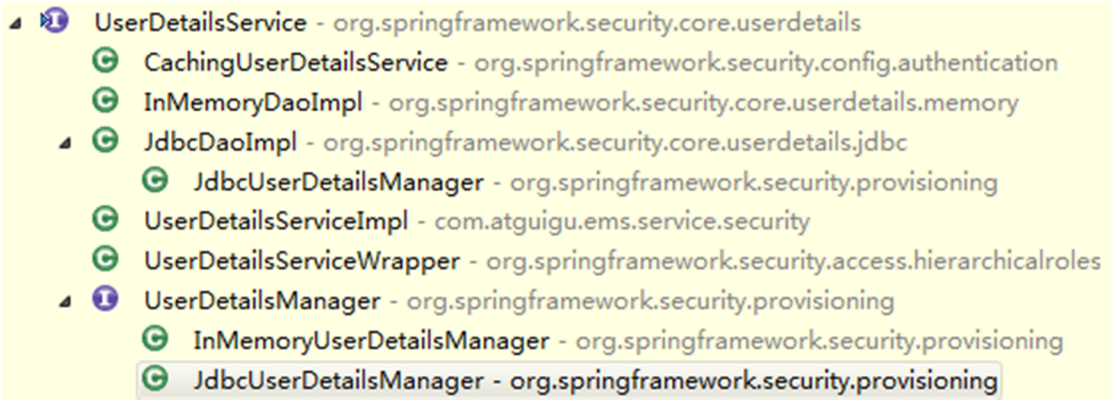
4. 把用户信息保存到数据库中的基本步骤：

1). 定义 UserDetailsService 接口的 Bean：在其中编写访问数据库的信息。具体获取的用户信息和用户所有的权限

①. public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException

②. 参照 SpringSecurity 的默认实现：

i. 通过 Eclipse 的 ctrl + T 可知 UserDetailsService 接口有如下的实现类;



ii. 经过初步分析，InMemoryUserDetailsManager 为采用配置文件实现的那个 UserDetailsService 的实现类

iii. 在 InMemoryUserDetailsManager 的 loadUserByUsername 方法的第一行打一个断点，使程序停下来，可以查看到返回的 UserDetails 对象的具体细节。

iv. 有 DEBUG 知道 User 对象的 authorities 中的元素的样子如下：

③. **实现 loadUserByUsername 方法：**

i. 提供 UserDetails 的实现类为 User

ii. User 有两个重载的构造器，暂时使用

public User(String username, String password, Collection<? extends GrantedAuthority> authorities)

其中 username 为用户名，password 为密码，authorities 为该用户所有的权限

iii. **可以通过如下方式来构建 Collection<? extends GrantedAuthority> authorities：**

```
//创建一个 GrantedAuthorityImpl 类型的集合对象，其中 GrantedAuthorityImpl 为
GrantedAuthority 接口的实现类
Collection<GrantedAuthorityImpl> authorities = new ArrayList<>();
//GrantedAuthorityImpl 可以通过带参数的构造器来创建，参数即为权限的名称
authorities.add(new GrantedAuthorityImpl("ROLE_ADMIN"));
authorities.add(new GrantedAuthorityImpl("ROLE_USER"));
```

④. 方法实现示例：

```
@Override
public UserDetails loadUserByUsername(String username)
        throws UsernameNotFoundException {
    UserDetails userDetails = null;

    String password = "123456";
    //创建一个 GrantedAuthorityImpl 类型的集合对象,其中 GrantedAuthorityImpl 为
    //GrantedAuthority 接口的实现类
    Collection<GrantedAuthorityImpl> authorities = new ArrayList<>();
    //GrantedAuthorityImpl 可以通过带参数的构造器来创建，参数即为权限的名称
    //authorities.add(new GrantedAuthorityImpl("ROLE_ADMIN"));
    authorities.add(new GrantedAuthorityImpl("ROLE_USER"));

    userDetails = new User(username, password, authorities);

    return userDetails;
}
```

2）. **把 UserDetailsService Bean 配置在 IOC 容器中**

```
<bean                                                    id="userDetailsService"
class="com.atguigu.springsecurity.service.UserDetailsServiceImpl"></bean>
```

3）. **使 <security:authentication-provider /> 的 user-service-ref 属性指向 2）配置的 Bean**

```
<security:authentication-manager>
    <security:authentication-provider user-service-ref="userDetailsService"/>
</security:authentication-manager>
```