

# 使用第三方系统的用户管理

## 1. 说明

第三方系统的用户管理能够包含 Smartbi 用户管理中的所有语义(用户组、角色、用户及其相应关系等),并且资源权限也希望由第三方系统统一管理,则可以屏蔽 Smartbi 的用户管理模块而使用第三方系统的用户管理作为统一用户管理平台。以下将介绍如何在外部系统中调用 Smartbi 用户管理模块。

## 2. 用户管理模块功能概述

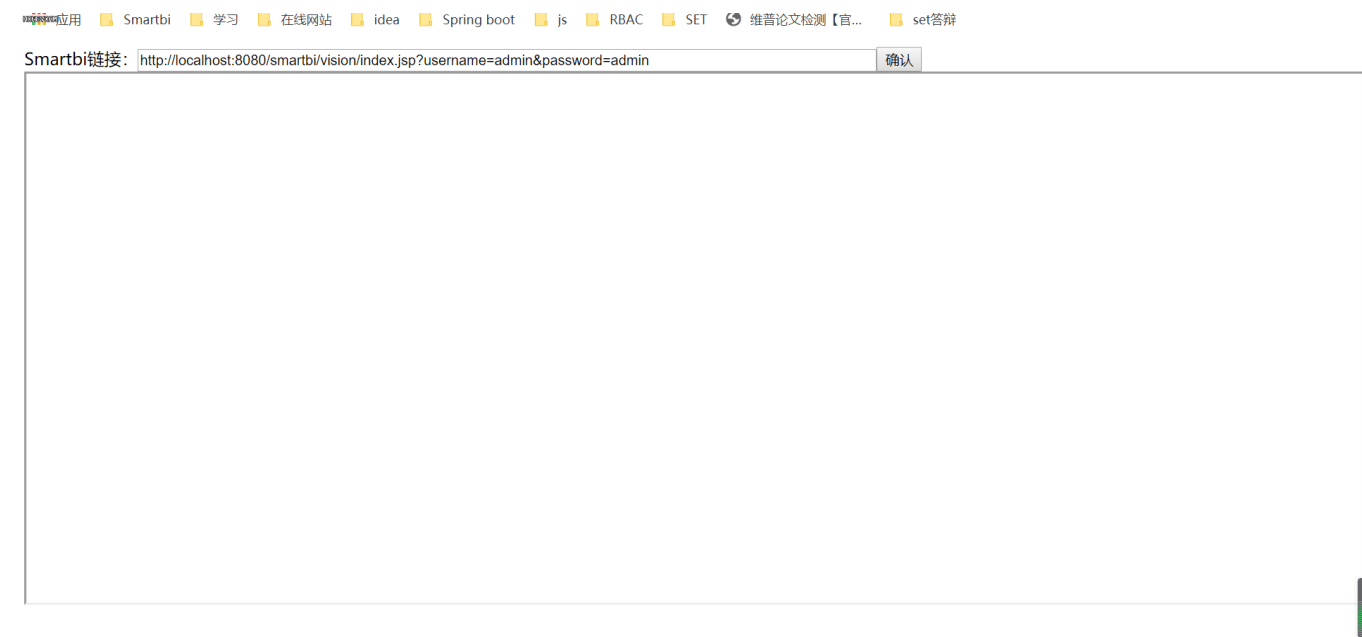
详情请参考wiki: [用户管理](#) 目录下的用户管理章节和权限管理章节。

- 1. 说明
- 2. 用户管理模块功能概述
- 3. 集成Smartbi用户管理模块到第三方系统
  - 3.1说明
  - 3.2集成用户属性
- 4. 同步Smartbi用户到第三方系统

## 3. 集成Smartbi用户管理模块到第三方系统

### 3.1说明

1. Smartbi 用户管理模块集成在Smartbi中,客户端通过IE访问,对用户、角色等进行管理。部署并启动Smartbi后,在第三方系统中通过本章后面的方法即可对用户管理模块进行访问。
2. 在第三方系统程序中通过IFrame访问 <http://server:port/smartbi/vision/index.jsp> 页面,并传递用户名、密码等参数过去,即可实现将用户管理模块嵌入到第三方系统中,效果如下。
3. 环境要求:
  - a. 软件要求: JDK 1.6 或以上, IE11;
  - b. 支持的服务器类型有: Tomcat 5.5、Weblogic 9.\*、WebSphere 6、Jboss 4.13 等。
  - c. 支持的数据库类型有: MySQL、SQL Server、Oracle、DB2、HSQL 等。



### 3.2集成用户属性

用户属性集成有三种方式,第一种是在使用iframe集成Smartbi的用户管理模块,直接在iframe中管理用户属性,第二种是直接操作用户属性对应的数据库表,第三种是通过服务器端SDK API管理用户属性。

#### 方式一：在集成的用户管理模块中增加用户扩展属性

根据以上步骤使用iframe在第三方系统中集成了用户管理模块后,可以在第三方系统中增加用户扩展属性,具体步骤如下:

- (1) 在系统运维 > 系统选项 > 用户管理设置 > 用户扩展属性 > 设置中



(2) 添加用户扩展属性



(3) 清空缓存, 重新进入用户管理, 对任意一个用户进行设置, 就可以看到刚刚添加的用户扩展属性

用户信息

角色列表

用户所属组

名称: \*

123

别名:

123

密码:

\*\*\*\*\*

确认密码:

\*\*\*\*\*

描述:

启用用户:

☒是

☐否

ces

手机号

报表ID

AAAA

是否强制修改密码

☐是

☒否

## 方式二：直接通过知识库以对用户扩展属性进行修改

用户扩展属性，定义在知识库的 `t_systemconfig` 表中，在其中 `c_key` 为 `USER_EXTEND_SETTING` 的行中，通过 `c_long` 列，以JSON形式定义。可通过下面的语句查询到该行数据：

```
SELECT * FROM t_systemconfig where c_key = "USER_EXTEND_SETTING";
```

如配置了前面的用户扩展属性后，查询结果可能为：

c_key	c_value	c_long
USER_EXTEND_SETTING		[{"name":"aTest","alias":"ces","typeS":"input","valueLength":"12"},{"name":"PhoneNumber","alias":"手机号","typeS":

修改 `c_long` 下的JSON即可对用户扩展属性进行修改。

## 方式三：通过服务器端SDK API对用户扩展属性进行读取

### （1）读取用户扩展属性结构

通过 `smartbi.sdk.service.systemconfig.SystemConfigService` 可以对用户扩展属性的结构进行读取：

```
ClientConnector conn = new ClientConnector("http://localhost:18080/smartbi");
conn.open("admin", "manager");
SystemConfigService service = new SystemConfigService(conn);
SystemConfig config = service.getSystemConfig("USER_EXTEND_SETTING");
System.out.println(config);
conn.close();
```

结果形如：

```
key=USER_EXTEND_SETTING;value=;longValue=[{"name":"1","alias":"","typeS":"","valueLength":"100"}]
```

### （2）修改、读取用户属性值

通过 `smartbi.sdk.service.user.UserManagerService` 可以对用户属性值进行读取、修改：

```
ClientConnector conn = new ClientConnector("http://localhost:18080/smartbi");
conn.open("admin", "admin");
UserManagerService service = new UserManagerService(conn);
service.updateUserAttribute("ADMIN", "1", "ExampleValue", "");
IExtensionAttribute attr = service.getUserAttribute("ADMIN", "1");
System.out.println(attr.getValue());
conn.close();
```

## 4. 同步Smartbi用户到第三方系统

### 实现步骤

1. 用户同步：通过额外的系统实现用户信息定时同步，有两种方式：
  - a. **方式一**：使用存储过程或 ETL 定时将 Smartbi 中的用户等信息同步到外部系统中，Smartbi 的用户相关的表结构可参考用户集成文档的同步数据库相关用户表。
  - b. **方式二**：Smartbi用户管理模块提供事件通知机制，当新增或修改用户等信息时，可通知第三方系统同时更新信息。这个要求第三方系统提供有二次开发接口，实现步骤如下：
    - i. 编写通知类\*\*UserManagerNotifier.java，如ThirdPartUserManagerNotifier.java，实现 IUserManagerNotifier 接口，在该类中调用第三方系统的用户管理 API 方法，实现用户操作的同步。IUserManagerNotifier 的作用是当 Smartbi 的用户、组、角色变化时通知继承该接口的类进行相关操作。具体接口方法请参考用户管理变更通知类 API 文档。
    - ii. 重新启动服务器，即可生效。
2. 用户验证：根据第三方系统的自身特征而定，可以使用其自身的用户验证，也可以通过Smartbi提供的二次开发接口使用 Smartbi 用户验证，详情请参考 [用户验证](#)。