

与第三方系统进行统一用户管理

用户管理模块是 Smartbi 产品线中一个独立的模块，Smartbi 系列的所有子产品均使用该模块进行用户管理和权限验证。

同时由于其独立性，该模块也可以独立出来给外部其它系统使用或实现与 Smartbi 产品的集成。下文主要介绍如何在外部系统中调用 Smartbi 用户管理模块。

文档目录:

- 1、用户管理模块功能概述
 - 1.1 说明
 - 1.2 权限体系
- 2、在第三方系统集成Smartbi用户管理模块
 - 2.1 说明
 - 2.2 用户属性集成
 - 2.2.1 扩展用户属性
 - 2.2.2 直接通过知识库以对用户扩展属性进行修改
 - 2.2.3 通过服务器端SDK API对用户扩展属性进行读取
 - 2.2.3.1 读取用户扩展属性结构
 - 2.2.3.2 修改、读取用户属性值
- 3、Smartbi用户同步到第三方系统
 - 3.1 实现步骤

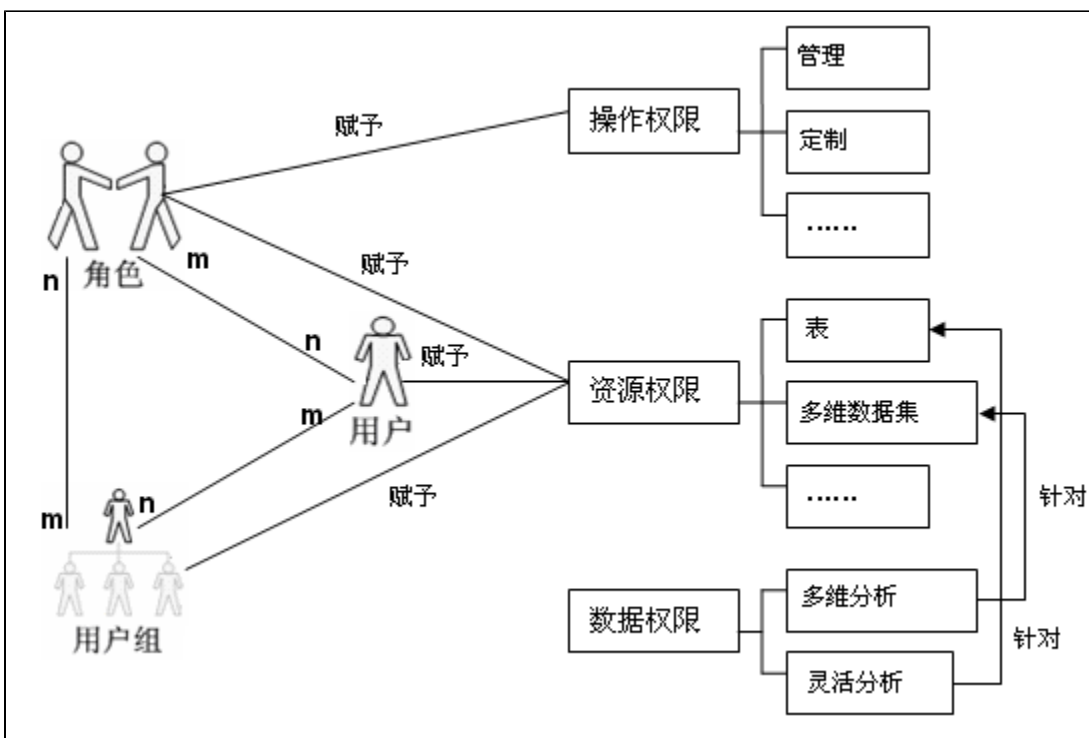
1、用户管理模块功能概述

1.1 说明

- 用户管理模块采用 B/S 架构实现。
 - 服务器端使用 Java 技术，可以在支持Servlet的Web 服务器中部署运行；
 - 浏览器端采用 Ajax 技术，除了拥有传统 B/S 架构应用的易于部署特点，还能够提供桌面应用，做到最佳用户体验。
- 用户管理模块的功能主要分四大部分：用户管理、组管理、角色管理、操作权限管理。
 - 用户管理：包括新建用户、编辑用户、删除用户、修改用户所属组、用户查找。用户通过修改用户所属组可以修改其所属组；用户查找可以对用户进行名称、别名搜索。
 - 组管理：包括新建组、编辑组、删除组。
 - 角色管理：包括新建角色、编辑角色、删除角色、修改角色用户、操作授权（对角色进行授权）。修改角色用户可以增加或减少某角色拥有的用户；操作授权可以把权限赋给某个角色，使角色拥有不同的权限。
 - 操作权限管理：包括新建权限、编辑权限、删除权限。

1.2 权限体系

- 用户权限体系如下图所示：



- Smartbi 用户管理模块通过对角色进行操作权限和资源权限授权，再将角色赋予给用户或组从而达到最终授权的目的。
- 用户可以属于多个组，拥有多个角色，默认继承组的角色。同时也可以将某个资源的权限赋给用户。
- Smartbi 用户管理模块定义了子系统概念，用户管理模块可以被多个系统集成，每个子系统具有各自的系统 ID，子系统的用户、组是所有系统公共的，而角色、操作权限每个系统不一样，即每个子系统拥有相同的组和用户，子系统可以根据需要定制本系统的角色和操作权限。一个用户可以登录所有的系统，但在每个系统中的角色和操作权限是不一样的，即用户或组在不同的子系统中可以拥有完全不同的、独立的角色和操作权限。

2、在第三方系统集成Smartbi用户管理模块

2.1 说明

- 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。
- 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。
- 环境要求：
 - 软件要求：JDK 1.6 或以上，IE11；
 - 支持的服务器类型有：Tomcat 5.5、Weblogic 9.*、WebSphere 6、Jboss 4.13 等。
 - 支持的数据库类型有：MySQL、SQL Server、Oracle、DB2、HSQL 等。

2.2 用户属性集成

2.2.1 扩展用户属性

1. 系统运维->系统选项->用户管理设置->用户扩展属性->设置

2. 添加用户扩展属性

用户扩展属性

属性名称	属性别名	控件类型	属性长度	删除
用户扩展属性1		文本框	100	

确定(O)

取消(C)

3. 清空缓存，重新进入用户管理，随便对一个用户进行设置，就可以见到刚刚添加的用户扩展属性

名称：

hb

别名：

华北

密码：

.....

确认密码：

.....

描述：

启用用户：

☒ 是

☐ 否

用户扩展属性1

是否强制修改密码

☐ 是

☒ 否

2.2.2 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":"用户扩展属性1","alias":"","typeS":"文本框","valueLength":"100"}]

修改 `c_long` 下的JSON即可对用户扩展属性进行修改。

2.2.3 通过服务器端SDK API对用户扩展属性进行读取

2.2.3.1 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"}]
```

2.2.3.2 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();
```

3、Smartbi用户同步到第三方系统

3.1 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：
 - **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 [与第三方系统用户集成](#) 中，[同步数据库相关用户表](#) 的章节。
 - **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：
 - a. 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。
 - b. 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。