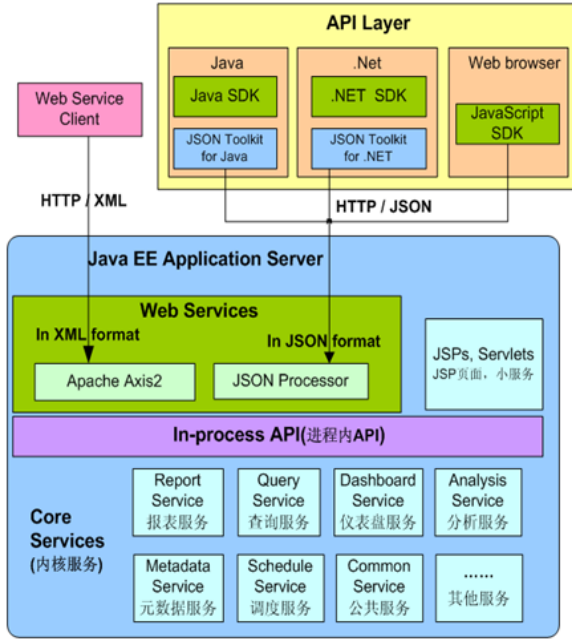


二次开发接口

Smartbi 提供三种形式的二次开发接口，以便实现更高要求的系统集成开发：

1. **服务器端SDK**：提供JAVA API 供外部系统调用实现集成要求。
2. **浏览器端SDK**：提供JavaScript API 供外部系统调用，满足非J2EE平台的客户环境（如.Net平台）实现与Smartbi 的集成要求。
3. **WebService服务接口**：通过 WebService 调用Smartbi API，满足非J2EE平台的客户环境（如.Net平台）实现与Smartbi 的集成要求。



文档目录：

- 1、服务器端SDK
 - 1.1 说明
 - 1.2 实现步骤
 - 1.3 调用示例
 - 1.4 注意事项
- 2、浏览器端SDK
 - 2.1 说明
 - 2.2 实现步骤
 - 2.3 调用示例
- 3、WebService服务接口
 - 3.1 说明
 - 3.2 注意事项
 - 3.3 示例

1、服务器端SDK

1.1 说明

服务器端 SDK 通过JAVA API 提供以下服务接口：

服务对象	描述
AnalysisReportService	提供 多维分析 相关操作功能。
BusinessViewService	提供 数据集 相关操作功能。
CatalogService	提供资源目录树的访问功能等。
SystemConfigService	提供系统设置相关操作功能。
OperationLogService	提供操作日志相关操作功能。
UserManagerService	提供用户相关功能，主要包括：读取/维护用户、组、角色等信息、为用户和组分配角色等。
ManageReportService	提供复杂报表相关操作功能（Deprecated）。

1.2 实现步骤

1. 假设Smartbi 服务器已经部署到应用服务器中，访问地址为：<http://localhost:18080/smartbi>。
2. 打开服务器部署文件smartbi.war，解压后将smartbi.war\WEB-INF\lib\目录下的**smartbi-Common.jar**、**smartbi-DAO.jar**、**smartbi-Framework.jar**、**smartbi-FreeQuery.jar**、**smartbi-SDK.jar**、**smartbi-UserManager.jar**、**ezmorph-0.8.1.jar**、**commons-logging-1.1.jar**、**commons-beanutils.jar**、**commons-collections-3.2.jar**、**commons-fileupload-1.2.1.jar**、**slf4j-api-1.6.4.jar**、**hibernate3-3.2.1.jar**、**log4j-1.2.13.jar** 等jar包加入到您的Java 项目的classpath 中去。

1.3 调用示例

```
String Smartbi_URL = "http://localhost:18080/smartbi";
ClientConnector conn = new ClientConnector(Smartbi_URL);
try {
    //
    boolean ret = conn.open("admin", "manager");
    if (ret) {
        //
        CatalogService catalogService = new CatalogService(conn);
        String elementId = "I2c9490741d4647ab011d4b92363f2061";
        String elementType = "SIMPLE_REPORT";
        boolean result = catalogService.isCatalogElementAccessible(elementId, elementType);
        System.out.println(result);

        //
        SimpleReportService simpleReportService = new SimpleReportService(conn);
        String reportId = "I2c9490741d2370a8011d2df3b1fd1fa0";
        Report report = simpleReportService.openReport(reportId);
        System.out.println(report.getCurrentReportName());

        //
        UserManagerService userManagerService = new UserManagerService(conn);
        String departmentId = "DEPARTMENT";
        IDepartment department = userManagerService.getDepartmentById(departmentId);
        System.out.println(department.getName());
    }
} catch (RemoteException e) {
    e.printStackTrace();
} finally {
    //
    conn.close();
}
```

1.4 注意事项

具体的方法以及帮助请参考[Java API](#)文档。

2、浏览器端SDK

2.1 说明

浏览器端SDK 通过JavaScript API 提供多个客户端组件供外部调用，满足非J2EE平台的客户环境（如.Net平台）实现与Smartbi 系统的集成要求。

2.2 实现步骤

1. 假设Smartbi 服务器已经部署到应用服务器中，访问地址为：<http://localhost:18080/smartbi/vision/>。
2. 打开服务器部署文件smartbi.war，解压后将\smartbi.war\vision\js\freequery\lang\目录下的JSLoader.js 文件复制到您的web路径下。

2.3 调用示例

```

<html>
<head>
    <title>javascript api </title>
    <script type="text/javascript" src="../../JSLoader.js"></script>
</head>
<body>
    <script type="text/javascript">
var BOF_UI_DEBUG = false;
var config = new Object();
config.baseUrl = "http://localhost:18080/smartbi/vision/"; //smartbiURL

// JS
var jsloader = new JSLoader(config);

//
var userService = jsloader.imports("bof.usermanager.UserService");

// userService.getInstance()
// UserManagerModule
var result = userService.getInstance().login("admin", "manager");
if (result) {
    alert("OK");
}
    </script>
</body>
</html>

```

3、WebService服务接口

3.1 说明

Smartbi 对外提供WebService 接口， 满足非J2EE平台的客户环境（如.Net平台） 实现与Smartbi 的集成要求。Smartbi提供以下WebService服务接口：

接口名称	接口说明	wsdl地址
AnalysisReportService	提供 多维分析 相关操作功能。	http://biserver:port/smartbi/vision/services/AnalysisReportService?wsdl
BusinessViewService	提供 数据集 相关操作功能。	http://biserver:port/smartbi/vision/services/BusinessViewService?wsdl
CatalogService	提供资源目录树的访问功能等。	http://biserver:port/smartbi/vision/services/CatalogService?wsdl
UserManagerService	提供用户、用户组、角色信息维护的相关接口。	http://biserver:port/smartbi/vision/services/UserManagerService?wsdl
ManageReportService	提供复杂报表相关操作功能（Deprecated）。	http://biserver:port/smartbi/vision/services/ManageReportService?wsdl

3.2 注意事项

具体的方法以及帮助请参考[WebService API](#)文档。

3.3 示例

- [.NET示例](#)
- [Java示例](#)