四川奔腾信息产业公司

**函数调用\_工具说明文档**

|  |  |
| --- | --- |
| 编 制 |  |
| 审 核 |  |
| 批 准 |  |

二〇〇九年七月二十三日

修改历史

|  |  |  |  |
| --- | --- | --- | --- |
| **日 期** | **作者/修订者** | **版本** | **修改原因** |
| 2009.7.14 | 廖朝敏 | 1.0 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1、目的 1](#_Toc236104168)

[2、名称与编号 1](#_Toc236104169)

[3、应用环境 1](#_Toc236104170)

[4、适用范围 1](#_Toc236104171)

[5、输入参数 1](#_Toc236104172)

[6、输出参数或结果 1](#_Toc236104173)

[7、涉及的其它工具 1](#_Toc236104174)

[8、使用说明 1](#_Toc236104175)

[9、使用示例 2](#_Toc236104176)

# 1、目的

提供函数调用的工具。

# 2、名称与编号

名称：函数调用

编号：

# 3、应用环境

本工具应用HTML，JSP等环境。

# 4、适用范围

# 5、输入参数

# 6、输出参数或结果

# 7、涉及的其它工具

暂无。

# 8、使用说明

1. 在dao.java中设置调用方法
2. 在daoImpl.java中写入实现方法。
3. 在service下XXXXXXService.java文件中，并在内写入调用方法。
4. 在service下XXXXXXServiceImpl.java文件中,并在内写入数据库接口Dao类方法，函数调用方法。
5. 在action中写入调用的方法
6. 在struts中配置文件

# 9、使用示例

1. 在dao.java中设置调用方法

/\*\*

\* 调用函数

\* @param functionName

\* @param params

\* @return Object

\*/

public Object callFunction(String functionName, List params);

1. 在daoImpl.java中写入实现方法。

public Object callFunction(String functionName, List params) {

return this.getPageTemplate().callFunction(functionName, params);

}

1. 在service下XXXXXXService.java文件中，并在内写入调用方法名。

如：

/\*\*

\* 得到调用

\* @return

\*/

public Object declareAuditing(Long declareId, Object Object);

1. 在service下XXXXXXServiceImpl.java文件中,并在内写入数据库接口Dao类方法，存储过程参数调用方法。

如：

/\*\*

\* <p>数据库Dao接口对象</p>

\*/

private Dao dao;

public int distributedRecords() {

UserValue userValue = UserValueProxy.getUserValue();

if(userValue != null){

//参数列表

List params = new ArrayList();

int paramlength = 8;

CallParameter callParameter = new CallParameter();

callParameter.setInOutTypes(CallParameter.TYPE\_IN);

callParameter.setTypes(Types.NUMERIC);

callParameter.setValue(Long.valueOf(userValue.getEntityId()));

params.add(callParameter);

callParameter = new CallParameter();

callParameter.setInOutTypes(CallParameter.TYPE\_IN);

callParameter.setTypes(Types.NUMERIC);

callParameter.setValue(Long.valueOf(userValue.getEntityId()));

params.add(callParameter);

callParameter = new CallParameter();

callParameter.setInOutTypes(CallParameter.TYPE\_IN);

callParameter.setTypes(Types.NUMERIC);

callParameter.setValue(Long.valueOf(userValue.getExhibitionId()));

params.add(callParameter);

.......

........

............

callParameter = new CallParameter();

callParameter.setInOutTypes(CallParameter.TYPE\_OUT);

callParameter.setTypes(Types.VARCHAR);

params.add(callParameter);

dao. callFunction ("{call PKG\_EBM.P\_EBM\_DISTRIBUTED\_RECORDS(?,?,?,?,?,?,?,?)}", params);

if(params == null || params.size() < paramlength){

throw new BtitException("情况统计异常");

}

callParameter = (CallParameter)params.get(paramlength - 1);

if(callParameter == null || callParameter.getValue() == null){

throw new BtitException("情况统计失败");

}

String sresult = (String)callParameter.getValue();

if(sresult != null && sresult.equals("-1")){

throw new BtitException("统计失败");

}

int result = ConvertUtil.stringToInt((String)callParameter.getValue());

return Object;

}

return null;

}