

风控防火墙二次开发说明

2018.7

目录

1. 风控防火墙二次开发	3
1.1 二次开发流程	3
1.2 准备环境	3
1.3 进行开发	3
1.4 功能发布	6

本文主要对风控防火墙二次开发进行介绍。为业务功能上二次开发提供扩展能力，支持委托厂家开发及公司自行开发。

通过阅读本文，您将不仅可以了解如何进行二次开发，也将能够运用本文所介绍的知识技巧和技巧，轻松构建属于自己的函数应用，文中将结合具体实例加以说明。

1. 风控防火墙二次开发

1.1 二次开发流程

主要步骤流程如下：

- 一、准备开发环境 ，准备需要的开发工具、编译环境
- 二、进行开发
- 三、功能发布

1.2 准备环境

开发工具:Eclipse

编译环境:Jdk1.7

1.3 进行开发

基于 Java 语言进行开发。Java 简便易用，不需要另外学习其他开发语言。

◇ 开发过程

- 1、建立 Java Project
- 2、新建类，创建要开发的函数应用 Java 类
- 3、实现代码业务逻辑

◇ Java 类方法说明

类中必须要有 exec 这个方法。

```
public HashMap<String, String> exec(HashMap<String, String> inputMap) ;
```

◇ 业务示例

- 规则需求：客户发生于非正常交易时间段（22:00-8:00）的交易笔数与全天总交易笔数之比大于 30%预警。

➤ 需求分析：

22:00-8:00 ,起始时间 22 点到截止时间 8 点 ,难点在于跨时间段。用正常的区间 ,
22 点<当前时间<8 点 ,无法实现 ,所以需要进行二次开发。

➤ 函数定义如下：

非正常交易时间段判断 , 实现指定时间区间判断 , 判断结果返回 (TRUE 或 FALSE)

输入参数：

编码 D1 ,需判断的时间(格式 : 年-月-日 时:分:秒 如 2018-03-21 12:46:16)

编码 STR1 , 起始时间 (格式 : 时 : 分 如 8:30)

编码 STR2 , 结束时间 (格式 : 时 : 分 如 22:30) ,

返回：

编码：RET_BOL , TRUE 或 FALSE 。

说明：编码由风控平台统一制定

➤ 代码如下：

```
import java.text.SimpleDateFormat;
import java.util.HashMap;

import org.apache.commons.lang3.time.FastDateFormat;

public class DateSpecify {

    public HashMap<String, String> exec(HashMap<String, String> inputMap) {
        HashMap<String, String> outputMap = new HashMap<String, String>();
        try {
            outputMap.put("RET_BOL", "");

            String d1 = inputMap.get("D1");//需判断的时间

            String start = inputMap.get("STR1");//起始时间

            String end = inputMap.get("STR2");//结束时间

            SimpleDateFormat dfs = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
```

```

FastDateFormat fm = FastDateFormat.getInstance("HH:mm");
String curDate = fm.format(dfs.parse(d1));

if (curDate.length() > 5 || curDate.length() < 4 || start.length() > 5 ||
start.length() < 4 || end.length() > 5 || end.length() < 4) {
    //System.out.println("input err: curDate=" + curDate + "|start=" + start +
"|end=" + end);
    return outputMap;
}
if (start.length() == 4)
    start = "0" + start;

if (end.length() == 4)
    end = "0" + end;

if (start.equals(end)) {
    outputMap.put("K", "0");
    outputMap.put("RET_BOL", "TRUE");//返回结果
} else if (start.compareTo(end) > 0) { // 起始时间大于 结束时间, 跨天 , 采
用或

    if (curDate.compareTo(start) >= 0 || curDate.compareTo(end) <= 0) {
        outputMap.put("K", "1");
        outputMap.put("RET_BOL", "TRUE");
    } else {
        outputMap.put("K", "2");
        outputMap.put("RET_BOL", "FALSE");
    }
} else {
    // 起始时间小于 结束时间, 采用且
    if (curDate.compareTo(start) >= 0 && curDate.compareTo(end) <= 0) {
        outputMap.put("K", "3");
        outputMap.put("RET_BOL", "TRUE");
    } else {
        outputMap.put("K", "4");
        outputMap.put("RET_BOL", "FALSE");
    }
}
//System.out.println("\n K=" + outputMap.get("K") + "|curDate=" + curDate +
"|start=" + start + "|end=" + end + "|RET=" + outputMap.get("RET"));
} catch (Exception e) {
    outputMap.put("RET_BOL", "");
    System.out.println(e.toString());
}
return outputMap;

```

```
    }  
}
```

1.4 功能发布

把.class 文件及类名，提交到风控平台，然后发布。发布后在新增指标时，就可以使用新开发的函数功能。