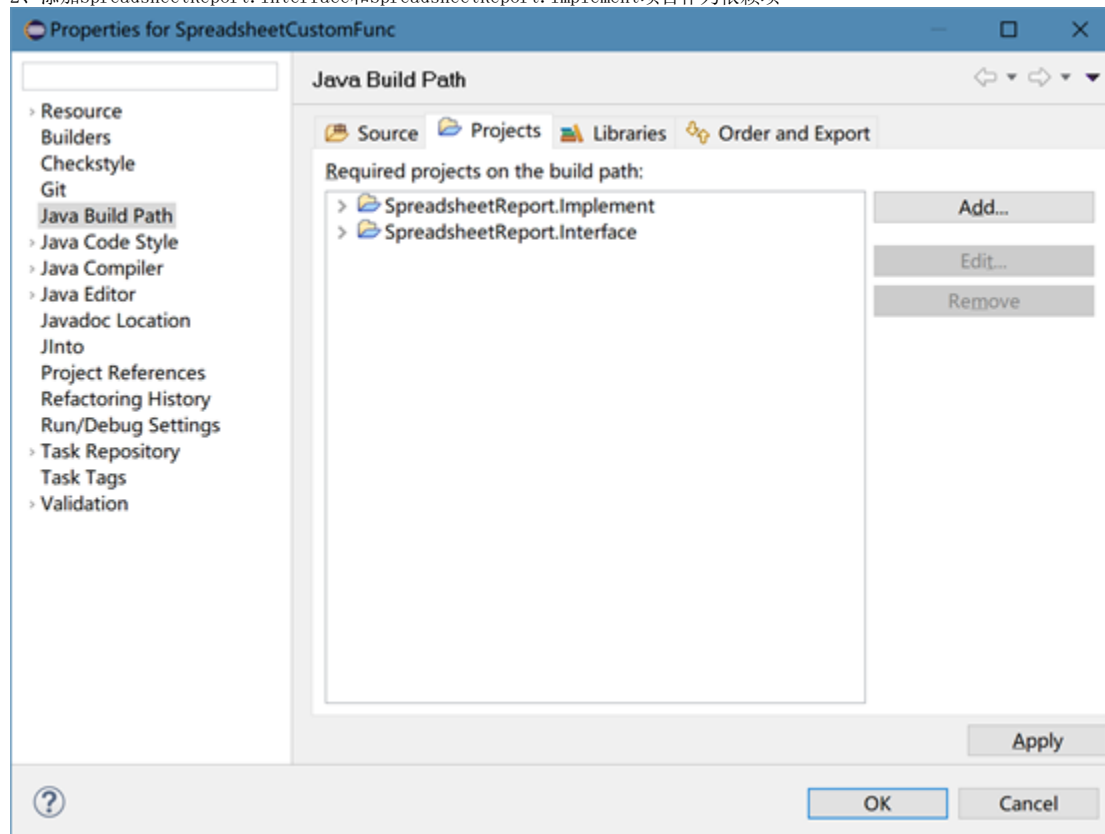


电子表格自定义函数

电子表格可以支持大部分的Excel内置函数，但是在某些场景下可能不能满足要求，需要定制开发实现，本文说明在电子表格中添加自定义函数。

环境准备

- 1、按wiki <https://history.wiki.smartbi.com.cn/pages/viewpage.action?pageId=27001774> 中的说明搭建一个新的扩展包开发项目
- 2、添加SpreadsheetReport.Interface和SpreadsheetReport.Implement项目作为依赖项



- 3、如果没有SpreadsheetReport源代码，也可以将smartbi.war/WEB-INF/lib中的所有jar文件作为项目的依赖项
- 4、在extensions.list文件中添加这个扩展包项目的src/web目录，使得Smartbi可以正常加载此扩展包
- 5、重启Smartbi服务器并在系统监控中的扩展包中确保扩展包已加载

函数实现

电子表格的自定义函数必须实现以下两个接口中的其中之一（不可以同时实现两个接口）

ICellFunction

```
package smartbi.spreadsheetreport.core.func;
public interface ICellFunction {
    /**
     *
     *
     * @param args
     *
     * @return
     */
    Object getData(Object[] args);
}
```

IGridFunction

```
package smartbi.spreadsheetreport.core.func;
import java.util.List;
public interface IGridFunction {
    /**
     *
     * @param args
     * @return
     */
    List<List<Object>> getDatas(Object[] args);
}
```

这两个函数的区别在于ICellFunction返回的是一个数据，IGridFunction返回一个二维数组的数据

函数使用

在Excel中使用函数SSR_ExecFunc和SSR_FillFuncData调用自定义函数，这两个函数的参数格式为：

- 1、 第一个参数是类的全名，当类的包名为smartbi.spreadsheetreport.core.func时可以只写类名
- 2、第二个及后面的所有参数值会传递到实现类中的args数组中：
 - 1) 支持常量
 - 2) 支持单元格引用
 - 3) 不支持任何公式，即SSR_ExecFunc("Percent", B4/3, C4)是不合法的，因为B4/3是一个公式
 - 4) 传递到args中的类型可能会改变为字符串，例如：SSR_FillFuncData("Matrix", 3, 4)在传入实现类时args的值为“3”，“4”而非整数类型，实现类需要自行进行数据类型的转换
 - 5) 不支持公式嵌套。以下情况不支持
 - SSR_ExecFunc("Percent", SSR_ExecFunc("Percent", 3, 4), C4)
 - D4单元格的公式为SSR_ExecFunc("Percent", B4, C4)，B4也是包含了函数SSR_ExecFunc
 - 6) 可以使用隐藏的单元格来实现简单的公式计算，例如C4使用公式B4/5，D4使用公式SSR_ExecFunc("Percent", C4, 10)，这样是可以支持的

实现示例

Percent

```
package smartbi.spreadsheetreport.core.func;
public class Percent implements ICellFunction {
    @Override
    public Object getData(Object[] args) {
        double d1 = convert(args[0]);
        double d2 = convert(args[1]);
        return d2 / d1 * 100;
    }

    private double convert(Object o) {
        if (o instanceof Number) {
            return ((Number) o).doubleValue();
        } else if (o instanceof String) {
            return Double.parseDouble(o.toString());
        } else {
            return Double.NaN;
        }
    }
}
```

这个类是将传入的两个参数进行了相除并乘100操作

Matrix

```
package smarthi.spreadsheetreport.core.func;
import java.util.ArrayList;
import java.util.List;
public class Matrix implements IGridFunction {
    @Override
    public List<List<Object>> getDatas(Object[] args) {
        int x = convert(args[0]);
        int y = convert(args[1]);
        List<List<Object>> result = new ArrayList<List<Object>>(x);
        for (int i = 0; i < x; i++) {
            List<Object> row = new ArrayList<Object>(y);
            for (int j = 0; j < y; j++) {
                row.add((i + 1) * (j + 1));
            }
            result.add(row);
        }
        return result;
    }
    private int convert(Object o) {
        if (o instanceof Number) {
            return ((Number) o).intValue();
        } else if (o instanceof String) {
            return Integer.parseInt(o.toString());
        } else {
            return 0;
        }
    }
}
```

这个类是示例填充一个N * M的矩形区域

使用示例

通过SSR_ExecFunc函数使用，Excel中的定义：

E16				
fx				
	A	B	C	D
1				
2				
3				
4		product(net_weight)	product(shelf_height)	=C4*3
5				=SSR_ExecFunc("Percent", 3, D4)

运行的效果：

6. 39	12. 60	37. 80	1260. 00
9. 60	18. 90	56. 70	1890. 00
4. 65	21. 90	65. 70	2190. 00
13. 80	18. 10	54. 30	1810. 00
	19. 00	57. 00	1900. 00
	20. 80	62. 40	2080. 00
	17. 50	52. 50	1750. 00
	8. 10	24. 30	810. 00
17. 00	16. 90	50. 70	1690. 00
	3. 46	10. 38	346. 00
	20. 40	61. 20	2040. 00
6. 97	11. 80	35. 40	1180. 00
5. 13	16. 10	48. 30	1610. 00
	18. 80	56. 40	1880. 00
7. 13	7. 32	21. 96	732. 00
11. 00	17. 20	51. 60	1720. 00
15. 50	17. 70	53. 10	1770. 00

通过SSR_FillFuncData函数使用，Excel中的定义：

B2		:	✕	✓	fx	=SSR_FillFuncData("Matrix", 3, 4)
	A		B			C
1						
2		⚠	=SSR_FillFuncData("Matrix", 3, 4)			
3						
4						
5						

执行的效果

🔄 ★ 📄 导出 🖨️		工作表：		Sheet1	Sheet2
		1	2	3	4
		2	4	6	8
		3	6	9	12

示例源码：[SpreadsheetCustomFunc.zip](#) 示例扩展包：[SpreadsheetCustomFunc.ext](#)