

15.BDF2-IMPORT

功能介绍

bdf2-import模块主要提供excel导入功能，可以将存放于Excel当中的数据批量导出到数据库（或者其它存储介质）。在某些项目当中，项目开发完成之后，接下来可能会面对大量基础数据录入的问题，而这些基础数据可能多数都以Excel形式存在。对于这种情况，我们可以利用BDF2当中提供的Excel数据导入功能，通过简单的模型定义，在不写一行代码的情况下，就可以将位于Excel中的数据导入到我们的数据库当中，从而实现大量、多种类型的基于Excel的基础数据的录入工作，bdf2-import功能模块具有以下特点：

- 提供可视化的Excel导入配置界面，可以在线对Excel文件数据进行预览、数据合法性校验、解析导入数据库；
- 可以同时支持Microsoft-Excel 97-2003(.xls)和Microsoft-Excel 2007+(.xlsx)；
- 通过实现指定的接口，可以对导入Excel数据进行自定义处理；

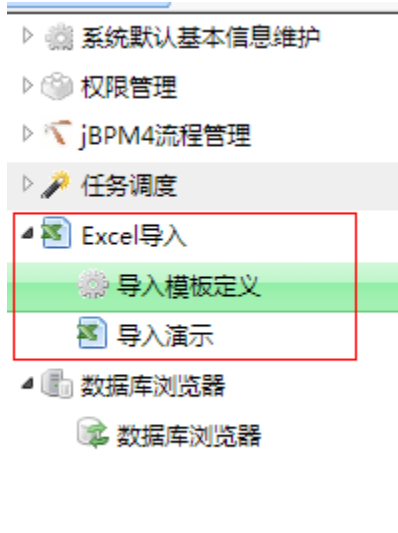
环境配置

要使用 bdf2-import模块，我们可以到nexus.bsdn.org上下载最新的bdf2-import模块的jar，或者可以到我们提供的[在线创建项目向导](#)中选择bdf2-import模块并下载即可；同样，如果您采用的是Maven来管理项目，那么只需要将bdf2-import模块的依赖信息加到我们的pom.xml当中即可：

bdf2-import模块所需要的依赖

```
<dependency>
  <groupId>com.bstek.bdf2</groupId>
  <artifactId> bdf2-import</artifactId>
  <version>2.0.0</version>
</dependency>
```

在bdf2-import模块当中，提供了提供可视化的Excel导入配置界面，所以，如果您在使用bdf2-import模块时，也同时使用了bdf2-core模块，那么可以在登录之后访问generate.system.menu.action这个用于初始化菜单的URL，创建好的导航菜单如下图所示：



导入模板定义

打开"导入模板定义"页面，添加一个导入方案，如下图所示。

| 方案编号 | 方案名称 | ExcelSheet页名称 | 数据源名称 | 物理表名称 | 物理表中文名称 | 物理表主键字段 | 主键类型 |
|------|------|---------------|-------|--------------|---------|---------|------|
| 0001 | 0001 | | mysql | bdf2_company | 公司表 | id_ | UUID |

| 名称 | Excel列号 | 物理表字段名称 | 单元格拦截器 |
|------|---------|---------|--------------------------|
| 1 名称 | 1 | name_ | bdf2.requiredInterceptor |
| 2 描述 | 2 | desc_ | bdf2.requiredInterceptor |

点击【添加】按钮，定义导入模板，如下图所示。

方案设置

方案编号:

方案名称:

ExcelSheet页名称:

处理方式: 系统默认处理类 自定义处理类

数据源名称:

物理表名称:

物理表中文名称:

物理表主键字段:

主键类型: 主键序列名称:

Excel导入处理类:

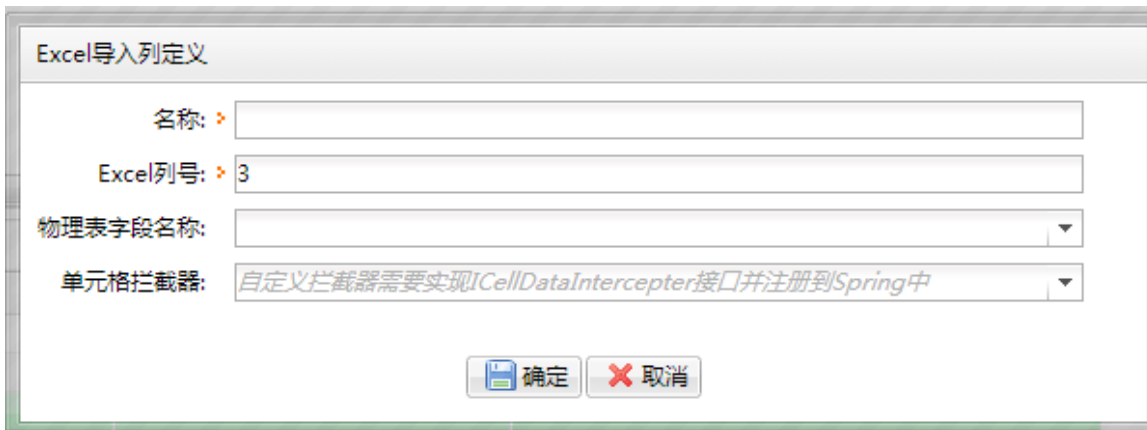
备注:

这里的需求注意的是导入模板的方案ID，这个ID首先需要唯一，然后尽量定义的有意义，因为我们后面在具体调用时要使用这个ID，其次需要注意的是如果导入数据的处理不采用“系统默认处理类”，而使用“自定义处理类”，那么我们写一个实现com.bstek.bdf2.importexcel.processor.IExcelProcessor接口的自定义类，并配置到spring环境当中，接口IExcelProcessor定义如下：

IExcelProcessor接口

```
package com.bstek.bdf2.importexcel.processor;
import com.bstek.bdf2.importexcel.model.ExcelDataWrapper;
/**
 * 要导入的Excel内容的处理类接口，如果自定义处理类，需要实现此接口
 *
 * @author matt.yao@bstek.com
 * @since 2.0
 */
public interface IExcelProcessor {
    public String getName();
    /**
     * 执行处理Excel内容的方法
     *
     * @param excelDataWrapper
     *      一个包装了Excel信息的集合
     * @return 导入处理成功返回的记录数
     * @throws Exception
     */
    public int execute(ExcelDataWrapper excelDataWrapper) throws Exception;
}
```

定义完方案之后，就可以为该定义信息添加具体与Excel对应的列信息了，如下图所示。



Excel导入列定义

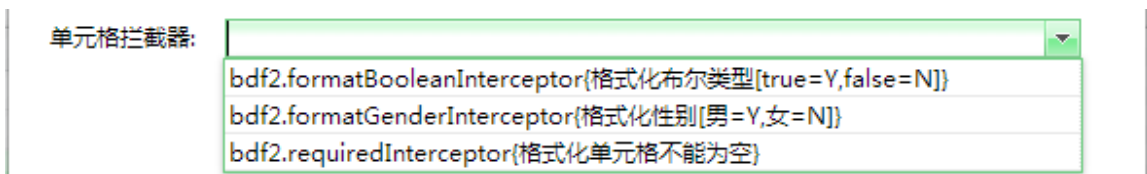
名称:

Excel列号:

物理表字段名称:

单元格拦截器:

同样在定义列信息时，对于列中数据的处理，还允许用户添加处理拦截类，这个拦截类的作用可以对导入的数据进行格式化（比如excel里定义的性别为男、女，存储到数据库中应该变为1和0等），或合法性验证（长度是否符合要求，是否为空等）。系统默认提供了三种拦截类，如下图所示：



单元格拦截器:

- bdf2.formatBooleanInterceptor(格式化布尔类型[true=Y,false=N])
- bdf2.formatGenderInterceptor(格式化性别[男=Y,女=N])
- bdf2.requiredInterceptor(格式化单元格不能为空)

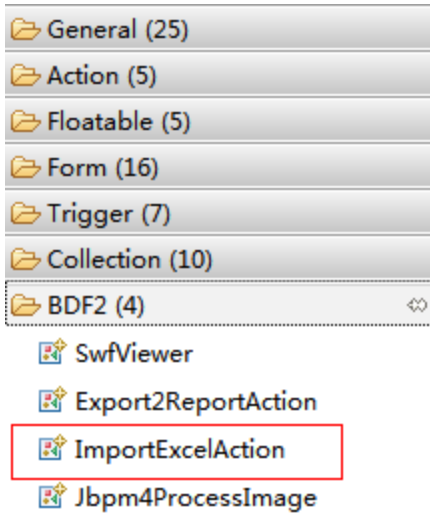
如果需要自定义拦截类，那么只需要定义一个类实现com.bstek.bdf2.importexcel.interceptor.ICellDataInterceptor接口，并配置到spring环境即可，例如系统中格式化性别的拦截类定义如下：

FormatGenderInterceptor

```
package com.bstek.bdf2.importexcel.interceptor.impl;
import org.springframework.stereotype.Service;
@Service("bdf2.formatGenderInterceptor")
public class FormatGenderInterceptor extends RequiredInterceptor {
    public Object execute(Object cellValue) throws Exception {
        super.execute(cellValue);
        if ("男".equals(cellValue)) {
            return "Y";
        } else {
            return "N";
        }
    }
    public String getName() {
        return "格式化性别[男=Y,女=N]";
    }
}
```

调用导入模板

定义好导入模板后，我们就可以在应用需要的地方调用这些定义好的excel模板，快速实现将Excel中的数据上传并导入到服务端。由于bdf2-import调用导入模板采用Dorado7组件方式实现，那么在创建配置完成项目后，需要启动项目工程更新Dorado7配置规则（在线更新），更新完Dorado7配置规则后，打开Dorado View文件，可以看到在Dorado7工具栏新添加了一个组件ImportExcelAction。如下图所示：



Auxiliary (6)

ImportExcelAction控件提供若干关键属性，如下表所示：

| 属性名 | 类型 | 默认值 | 描述 |
|----------------|---------|-------|---------------------------------------|
| excelModelId | String | 空 | 定义导入模板时指定的方案编号 |
| startRow | int | 空 | excel的开始行，如果不指定默认值，默认第一行开始解析 |
| endRow | int | 空 | excel的结束行，如果不指定默认值，系统自动判断excel的结束行 |
| showImportData | boolean | true | 是否在线预览导入的数据，默认true |
| bigData | boolean | false | 如果数据量比较大，可以设置为true，解析的excel单元格值类型为字符串 |

在view中添加一个ImportExcelAction控件，设置对应的模板方案编号，调用示例代码如下：

```
var importExcelAction=view.get("#importExcelActionDemo");
importExcelAction.set("excelModelId","0001");
importExcelAction.execute(function(arg){
    dorado.MessageBox.alert("成功解析导入[" + arg + "]数据！");
});
```

整个调用界面效果如下面几张图所示：

