目录

[一、类结构图 2](#_Toc764684411)

[1.1 类结构图 2](#_Toc1508752029)

[1.2 ExcelFactoryConfig配置项(已过期) 2](#_Toc108447627)

[1.3 ExcelFactoryConfigInner(3.25更新) 3](#_Toc1613134333)

[1.4 ExcelFactory主体流程图 4](#_Toc2115175003)

[二、代码说明 4](#_Toc301982983)

[2.1 核心类说明 4](#_Toc924137420)

[2.1.1 Column注解 4](#_Toc1375882836)

[2.1.2 Excel注解 5](#_Toc358913756)

[2.1.3 ColumnWrap类 5](#_Toc2129416316)

[2.1.4 Validator类 6](#_Toc1285045757)

[2.1.5 Converter接口说明 8](#_Toc521000020)

[2.2 异常体系 9](#_Toc1156507321)

[2.2.1 异常体系继承图 9](#_Toc544055050)

[2.2.2 异常继承关系说明 9](#_Toc2095340071)

[三、相关拓展点 11](#_Toc1943729791)

[3.1 如果存在一个Bean有多个解析版本的情况 11](#_Toc745359173)

[3.1.1 多版本实现的方式 12](#_Toc979507660)

[3.1.2 多版本实现的原理 12](#_Toc2123087365)

[3.2 动态表头的Excel 12](#_Toc141065003)

[3.2.1 核心类说明 12](#_Toc57559133)

[3.3 配合Excel注解在Controller层中将Excel转换成Bean 14](#_Toc1028707181)

[3.3.1 springmvc扩展点简单说明 14](#_Toc90749070)

[3.3.2 @ExcelToBean注解说明 14](#_Toc506230120)

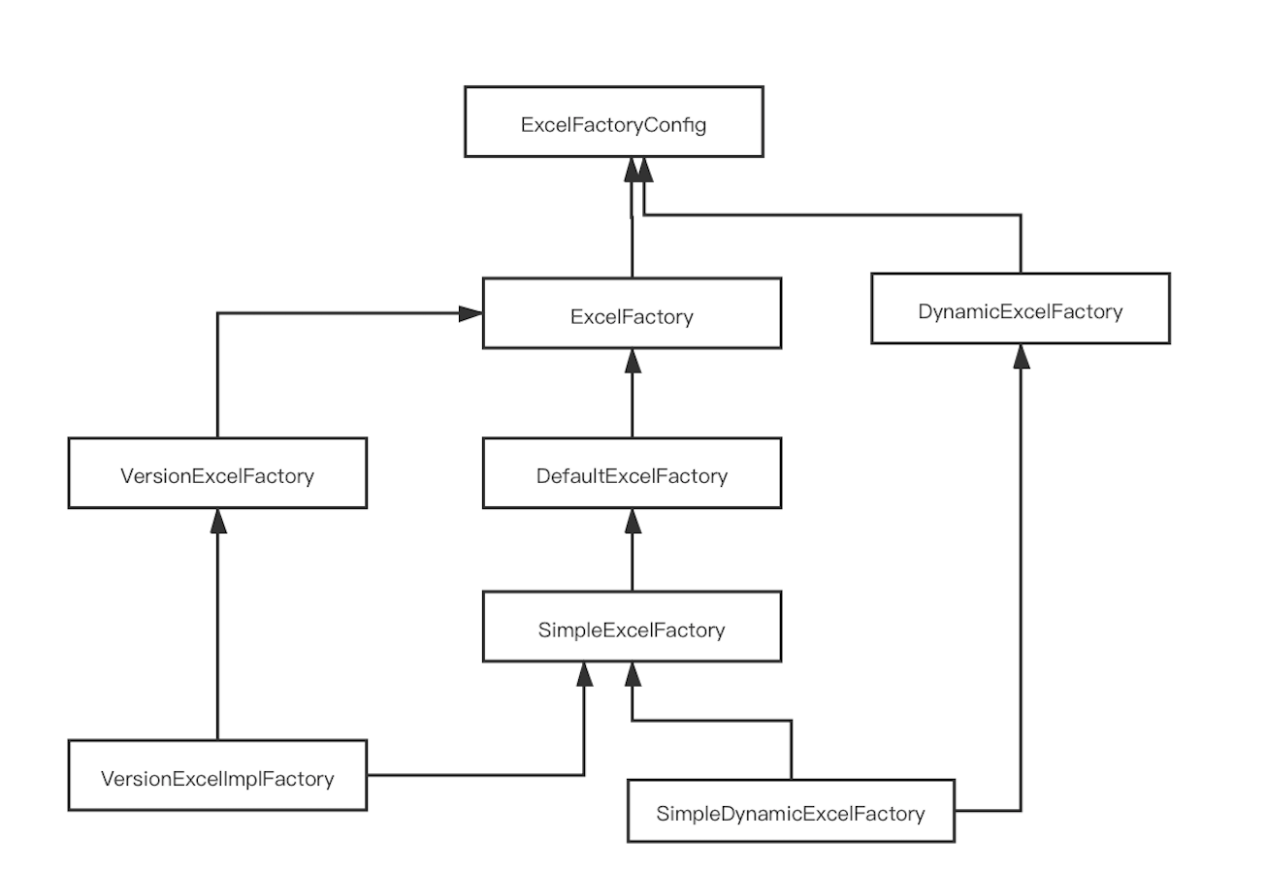
[3.3.3 使用方法 14](#_Toc2026901073)

[3.4 举例行过滤(3.25新增功能) 15](#_Toc593241550)

[3.5 全异常捕获(3.25新增) 15](#_Toc1991641476)

# 一、类结构图

## 1.1 类结构图



## 1.2 ExcelFactoryConfig配置项(已过期)



此配置项只提供了是否将空白单元格解析成空字符串，如果为true，则对应字段为null。

*#3.25更新*

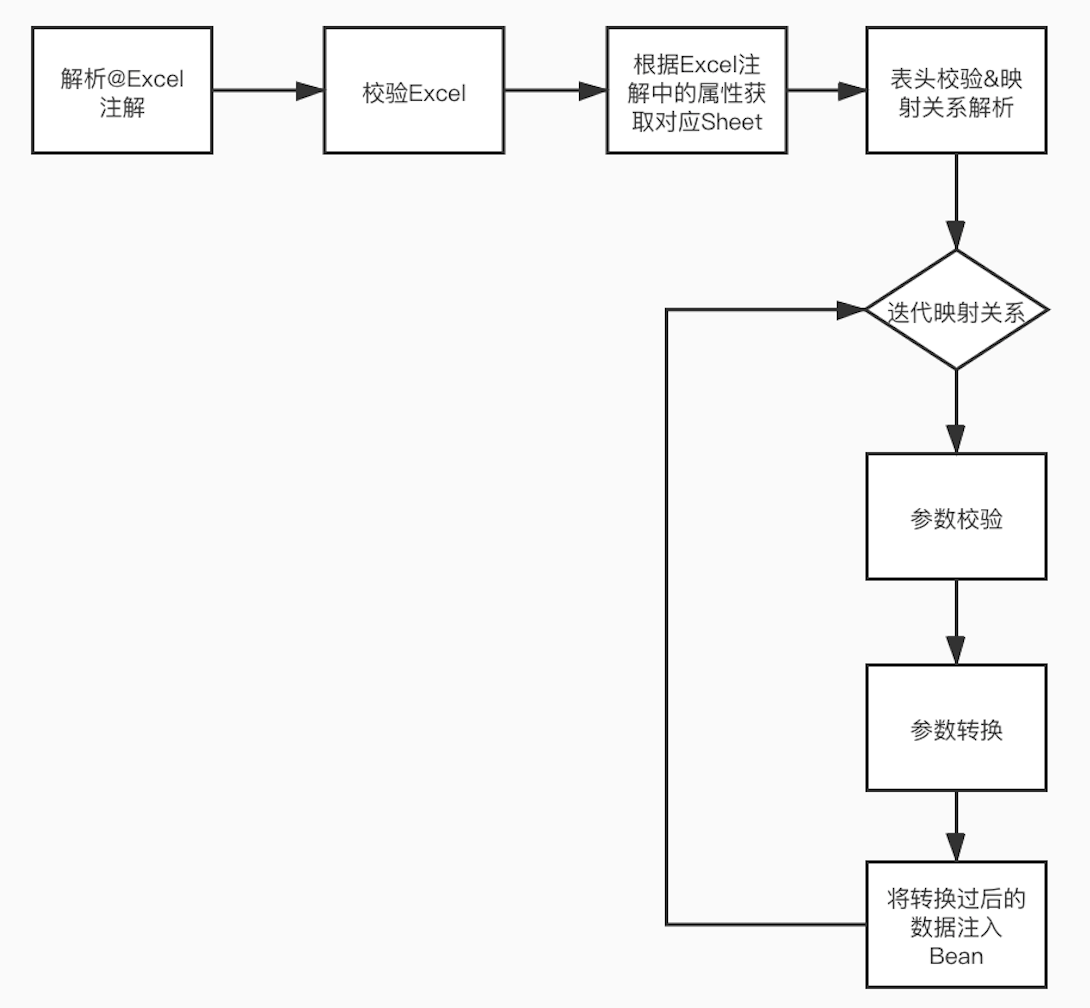
*此接口已经被标注为过期，请用1.3中ExcelFactoryConfigInner*

## 1.3 ExcelFactoryConfigInner(3.25更新)



工厂的配置信息由1.2中的接口实现改成配置类聚合，目前提供了三种配置项目(原有的接口模式也保留，只是不推荐使用，以后新增的配置项皆走ExcelFactoryConfigInner)

## 1.4 ExcelFactory主体流程图



# 二、代码说明

## 2.1 核心类说明

### 2.1.1 Column注解

Column注解的参数名意义如下

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 默认值 |
| version | 多版本控制 | -1 |
| position | 字段在excel中对应的位置(暂未使用) | -1 |
| headerName | 对应列表头的名称 | 无 |
| setter | 自定义的setter方法 | “”,代表将自动使用setXxx()的方法 |
| getter | 自定义的getter方法 | “”,代表将自动使用getXxx()的方法 |
| required | 此表头是否必须存在 | true |
| converter | 自定义的转换器，如果没有设置将使用Spring框架提供的ConversionService尝试进行转换 | EmptyConverter |
| valid | 校验器，支持多个校验器同时校验，校验的顺序即为从左到右，如果不设置则不进行校验 | {NopValidator} |
| argsValid(新增) | 新增带参数的校验器，与valid同时使用时会使得valid失效 | [] |
| format | 格式化类，在generateExcel方法中使用，对bean进行format，在文件上传时不起作用 | DefaultFormatter |

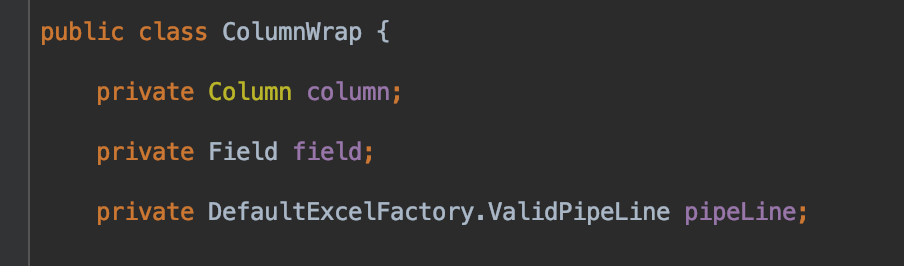
### 2.1.2 Excel注解

Excel注解的参数名意义如下

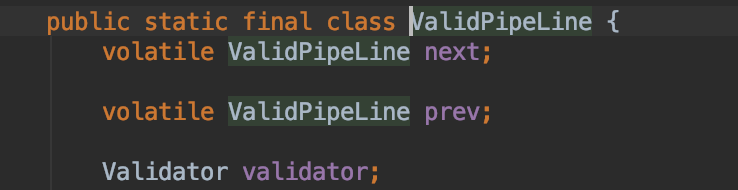
|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 默认值 |
| offset | 表头的偏移量 | 0 |
| sheet | excel表格的sheet编号 | 0 |
| sheetName | sheet名称，优先级高于sheet | “” |

### 2.1.3 ColumnWrap类

ColumnWrap类有3个成员变量，分别是1）Column 2）Field 3）ValidPipeline



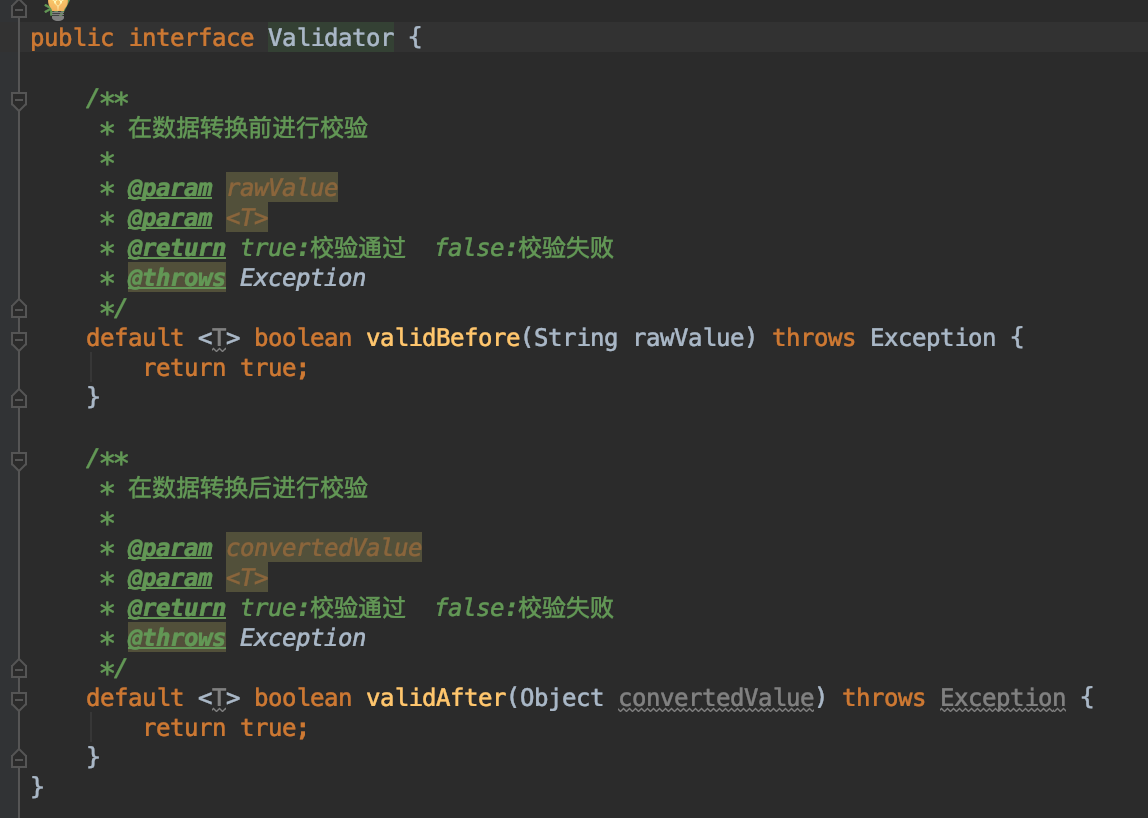
一个ColumnWrap实例代表了excel中某一列于目标bean中的某一个field的对应关系，pipeline代表根据Column中valid属性中的校验器数组，为了后期使用的方便，预先解析出来，其内部结构为链表。



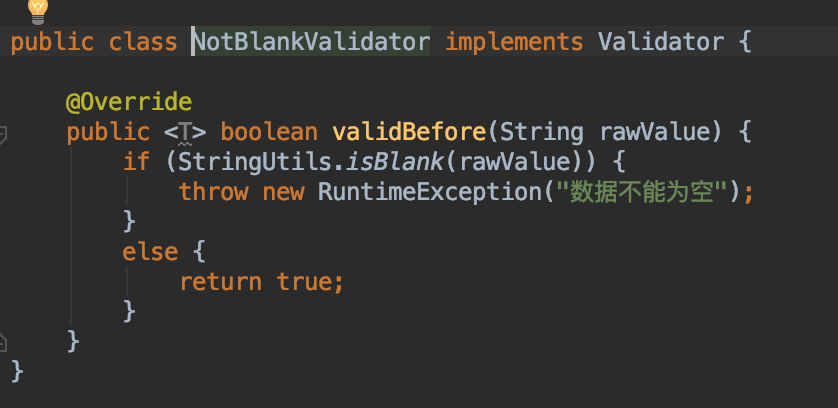
根据传入的excel解析出来的多个ColumnWrap组成数组作为ExcelBeanMetaData的成员属性，用于解析excel的唯一依据。

### 2.1.4 Validator类

Validator接口定义了两个方法：



分别代表对原始数据转换前的校验以及数据转换后的校验，如下举一个最简单的非空校验



如果校验失败并且想要将错误信息抛出给上级作为展示或其他的用途，需要直接抛出一个异常，并且将错误提示信息传入构造体中。

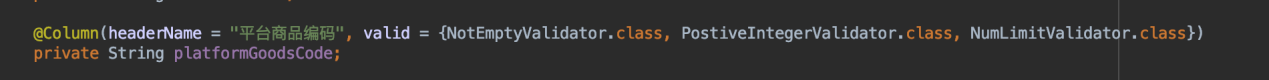
需要注意的是：

1. 如果使用@Valid注解进行使用时，校验器需要提供一个空构造器。
2. 如果使用@ArgsValidators注解进行使用时，校验器需要根据配置提供相对应的有参构造方法。

#### 2.1.4.1 校验器的组合使用

校验器可以进行多个及联使用，比如某一个字段的校验要求是：1.不可以为空 2.必须是正整数，并且小于100。

那么可以对三个校验器进行拼装完成校验，PostiveIntegerValidator校验器只关注数据是否为正整数，而不用去关注数据是否为空，因为校验器是按照顺序进行校验的，一旦非空校验不通过，后面的校验器不会去执行，也可以选择一个校验器同时做以上三个判断，分开的话可以更加灵活并且容易复用。



#### 2.1.4.2 校验器隐式传参

如果校验器需要一个在运行时才能知道的参数，例如需要根据用户选择的店铺进行校验时，可以使用隐式传参数。

隐式传参数的相关功能由ParameterPassHelp类实现，实现原理即用ThreadLocal保存参数，根据特定KEY获取相对应的参数，KEY与校验器指定的KEY保持一致即可。

**切记！！如果特殊情况下进行了线程切换，需要手动将参数再传递一遍！！！**

#### 2.1.4.3 带入参的校验器

**截屏2020-08-04 下午9.46.18**

校验器需要根据配置提供相对应的有参构造方法。

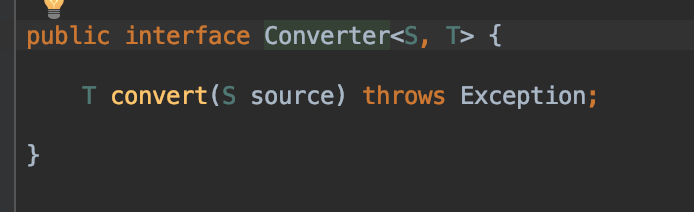
实现带入参的校验器过程中新引入了@ArgsValidators注解，其有三个参数：

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 默认值 |
| validator | 校验器的class | default |
| args | 校验器构造器的参数(多个) | {} |
| argsClass | 校验器构造器的参数类型 | {} |

其中args为字符串类型的参数数组，argsClass为参数的class类型，args的长度必须和argsClass的长度相等，并且是能转换的，否则将报错，例如无法将“abc”转换成Integer类型(自己定义了转换方式的除外)。

### 2.1.5 Converter接口说明

Converter接口定义了一个方法：

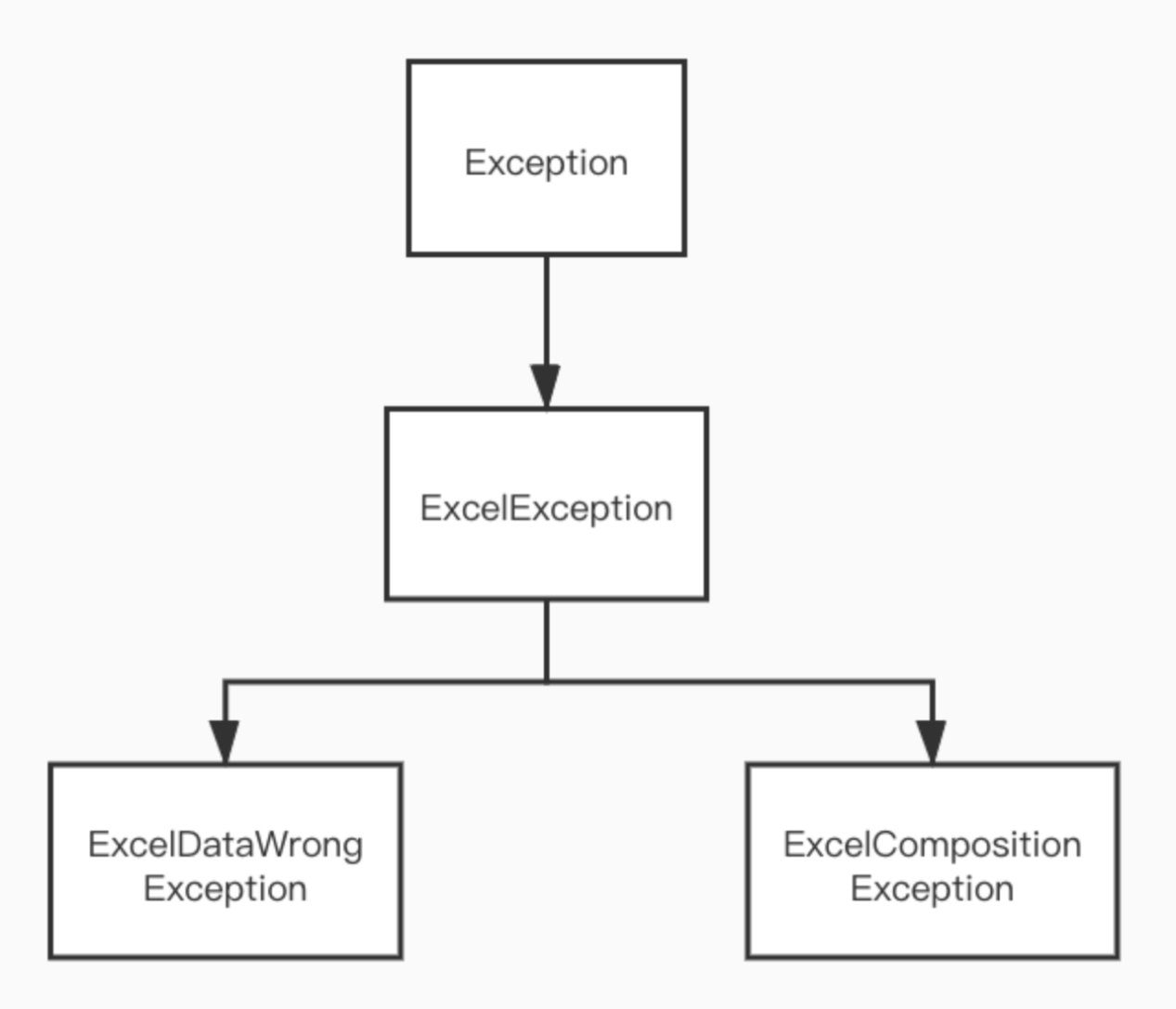


通过自定义接口实现可以将cell中的数据转换成所需要的格式，如果@Column注解中有配置对应的Converter类，则会使用此Converter进行参数转换，如果没有配置，则默认会使用Spring提供的ConversionService根据所需要的字段类型进行参数的自适应转换。

通过常见的Converter实现类，可以做到固定excel的某一列、去除cell中所有空格、自定义枚举类型转义等功能。

## 2.2 异常体系

### 2.2.1 异常体系继承图



### 2.2.2 异常继承关系说明

ExcelException是抽象类，其内部展示没有字段以及重写方法，目前只是扮演一个tag的角色，有利于业务层去捕获框架中的异常。

框架中的异常类，继承于ExcelException，Excel解析过程中的异常信息会被捕获并且封装成ExcelDataWrongException，其有三个成员变量：

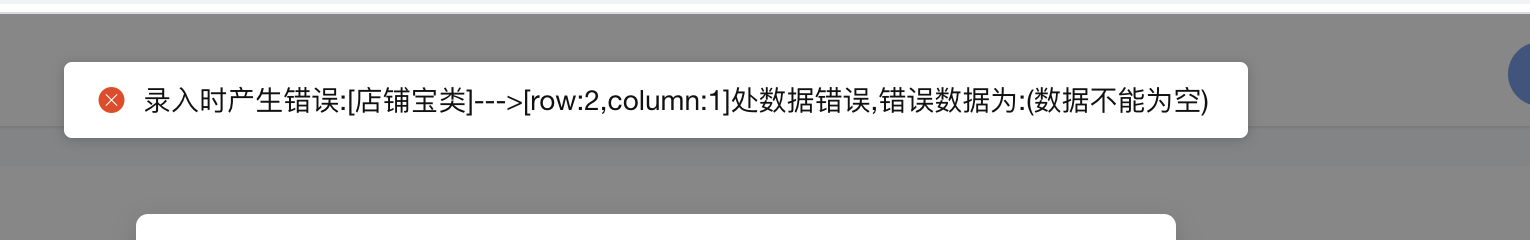


ExcelPos代表解析哪一个单元格时发生了异常，其内部记录了单元格的行和列信息，invalidData代表了发生异常的单元格的数据，至于发生的异常(包括converter和validator中的异常信息)存放于Throwable中的detailMessage中。

ExcelDataWrongException定义了默认的错误展示格式，其格式为：

**[sheet名称]--->[row:1,column:1]处数据错误,错误数据为:123(数据不能大于100)**

括号内的文字是异常信息(如果有异常信息，没有则不展示)。如果需要展示自定义的信息格式，可以重新写一个异常类并继承ExcelDataWrongException类后重新toString方法，或者在业务代码中捕获ExcelDataWrongException后重新封装后抛出。



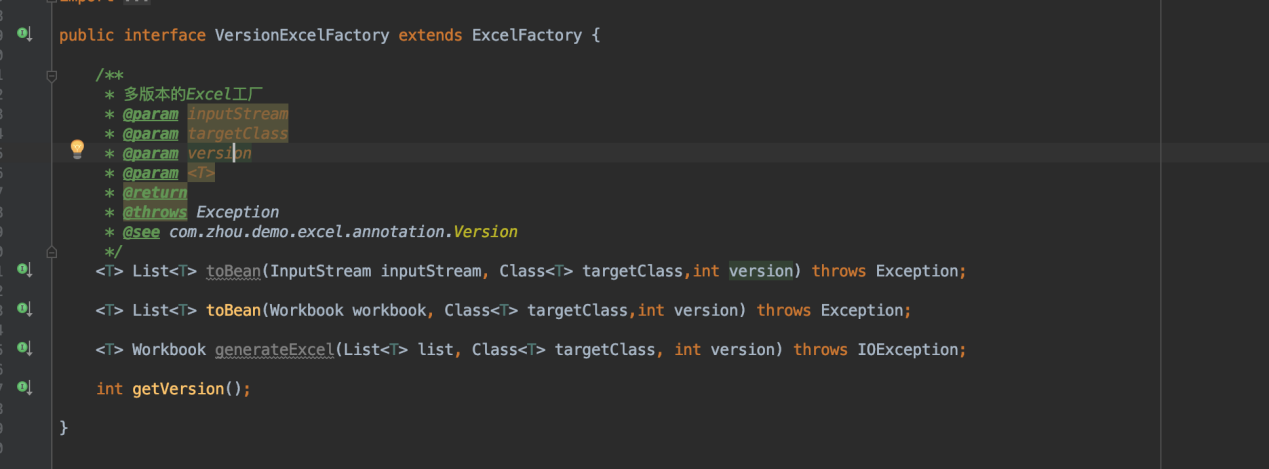
ExcelCompositionException是一个聚合异常，其内部封装了一个存放ExcelDataWrongException的list集合。

****

# 三、相关拓展点

## 3.1 如果存在一个Bean有多个解析版本的情况

**多版本的API位于VersionExcelFactory中，相比于ExcelFactory提供的API，主要是增加了version字段。**

****

### 3.1.1 多版本实现的方式

****

### 3.1.2 多版本实现的原理

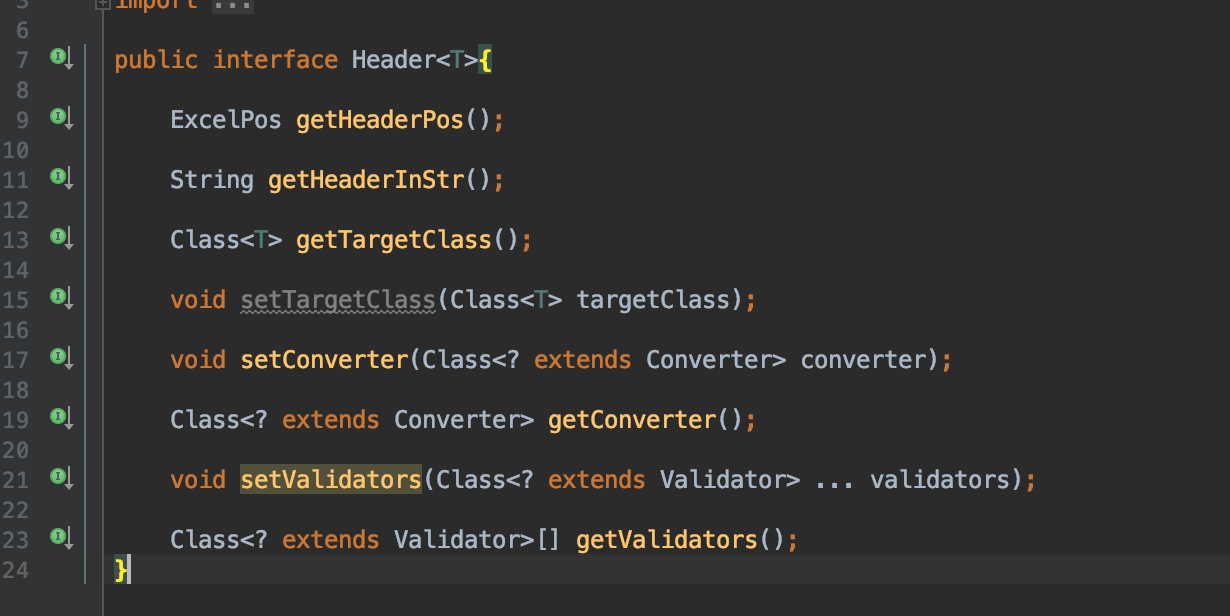
多版本实现的方式非常简单，VersionExcelFactoryImpl重写了resolveMapping方法，其内部新增了对@Version注解的解析，根据提供的版本号选择不同的校验&转换规则，在流程图中只是影响了「映射关系解析」这一步，对于后续的流程是无感知的，也可以根据这一特性自定义解析规则，来扩展框架。

## 3.2 动态表头的Excel

### 3.2.1 核心类说明

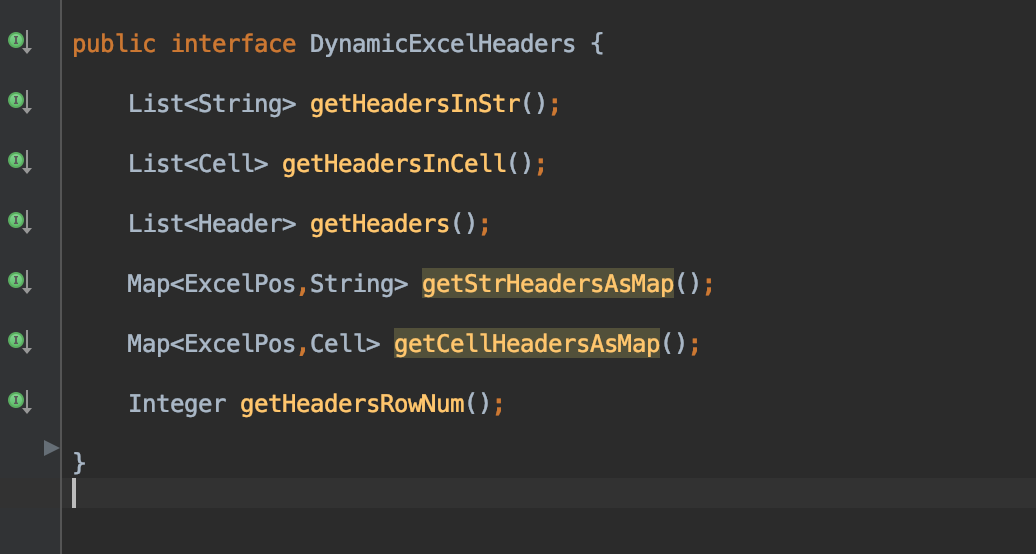
#### 3.2.1.1 Header

由于动态表头的Excel解析时表头不固定，所以无法和固定表头的Excel解析一样通过字段注解进行解析，因此将动态Excel的每一列抽象成Header对象，Header对象也和静态Bean一样，也提供了校验器和转换器，只是无法通过声明式配置，而是需要通过编程方式配置。

****

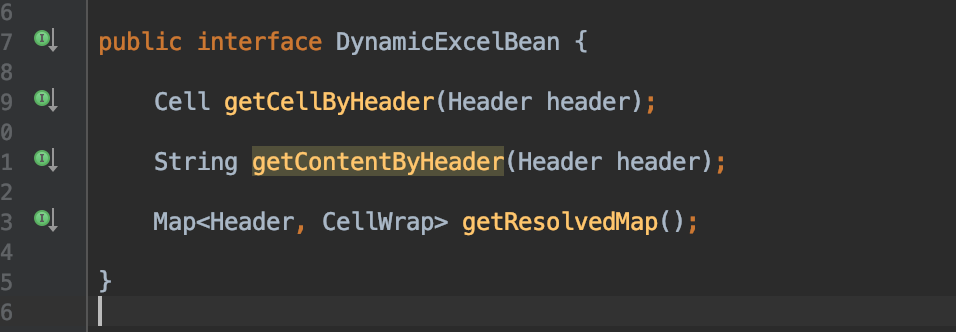
#### 3.2.1.2 DynamicExcelHeaders

**DynamicExcelHeaders可以认为是Header的聚合类，在将Header打包的同时提供了一组针对Header进行操作和管理的API。**

****

#### 3.2.1.3 DynamicExcelBean

**与Header类相似，DynamicExcelBean也是一种抽象的数据形式，一个DynamicExcelBean代表的意义可以理解为一行数据的聚合，可以通过Header(可以理解为Bean中的字段名)去找到对应的列数据。**

****

## 3.3 配合Excel注解在Controller层中将Excel转换成Bean

### 3.3.1 springmvc扩展点简单说明

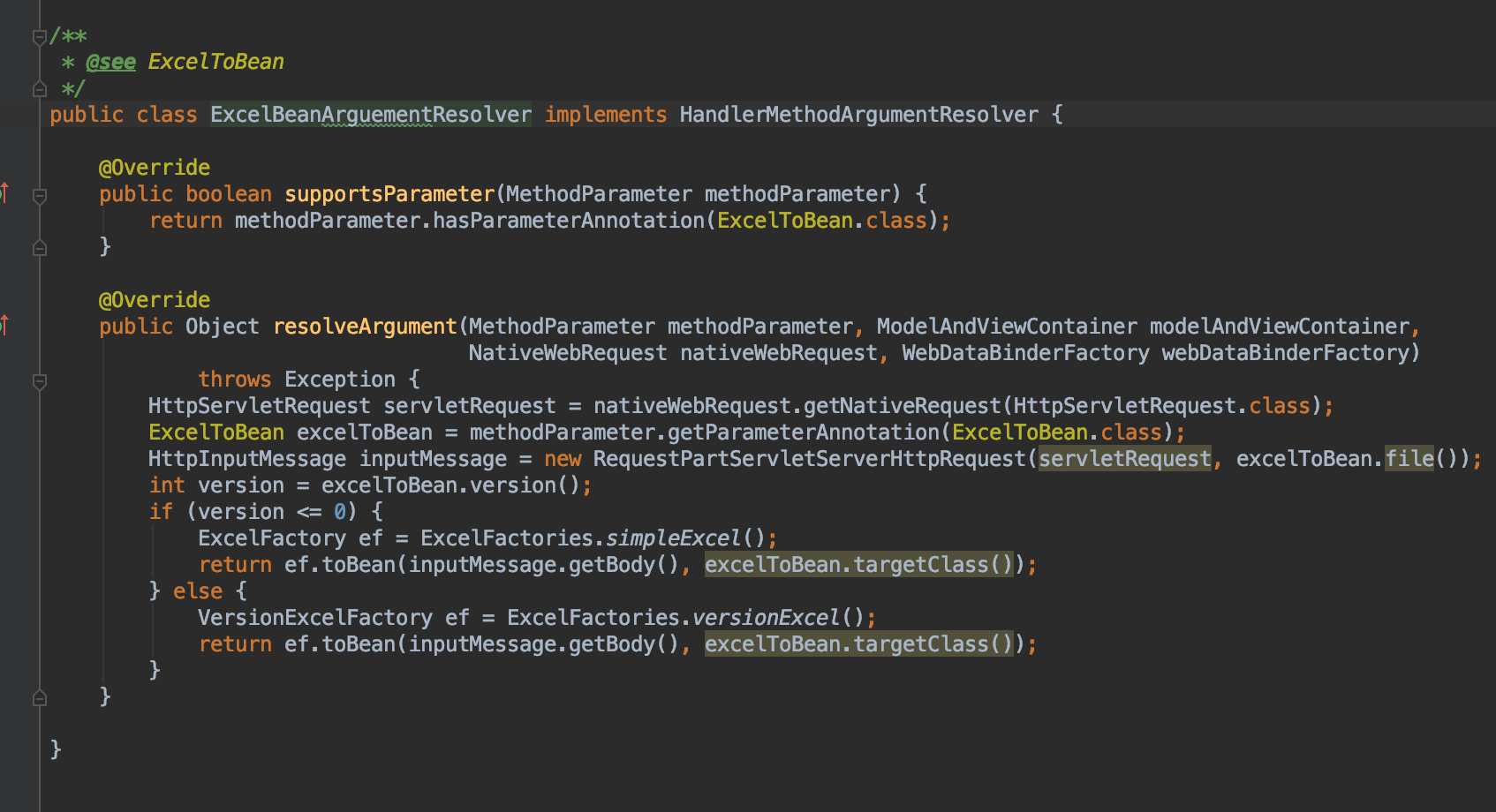
**基于RequestMapping注解注册的url，其controller层的参数解析是基于HandlerMethodArgumentResolver接口来实现的，将Excel转换成Bean需要基于此接口拓展出处理Excel的自定义实现类。**

### 3.3.2 @ExcelToBean注解说明

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 默认值 |
| targetClass | 目标Bean | 无，必填项 |
| version | 版本号，如果是非多版本可以不用填写 | -1，默认无版本 |
| file | 表单提交文件时，分配给文件的key值 | 无，必填项 |

### 3.3.3 使用方法

解析Excel转换成Bean的源码位于ExcelBeanArguementResolver中，使用时像@RequestBody一样打在入参上即可，注意入参必须是List对象，否则注入会失败。



## 3.4 举例行过滤(3.25新增功能)

举例行出现在表头的下一行，并且单元格都以“例”开头，则判断此行是举例行并且略过，用户可以手动删除此举例行，如果不删除也不会像常规单元格一样进行校验、转换。

可在ExcelFactoryConfigInner配置类中设定filterExample=true/false来开启/关闭这项功能。

## 3.5 全异常捕获(3.25新增)

框架整合了全异常捕获，可以一次性返回所有的异常项目，所有的异常信息会放置于ExcelCompositionException中，此异常中包含多个ExcelDataWrongException。

