# Mutils-spring-boot-starter使用手册

说明:Mutils-spring-boot-starter是基于jdk8和SpringBoot2.0.0.RELEASE的一套API集成框架，其中提供了支付宝、微信支付(jsapi、app、小程序)、点我达、移客通、excel、文件上传、快递100、移客通API的二次封装，提供了一系列便捷调用方法。集成springboot特有的自动化配置，将一系列的签名、配置文件、所需参数生成对应实体类并且在yml或properties中以快捷方式展现。每个模块中可以使用的功能，都是以xxxxxFunctions进行命名。以下简称框架。

可在github中下载最新的代码,地址如下:

<https://github.com/mintonzhang/mutils-spring-boot-starter.git>

1. **使用前提:**

1.jdk不能低于1.8

2.springboot不能低于2.0.0.RELEASE

1. **开始使用**
   1. **引入pom依赖**

**注意事项:**使用在maven中心仓库搜索groupid(cn.minsin)即可搜索到。新项目都尽量使用最新版本。**目前最新版本为0.2.0**

<!-- https://mvnrepository.com/artifact/cn.minsin/mutils-spring-boot-starter -->

<dependency>

<groupId>cn.minsin</groupId>

<artifactId>mutils-spring-boot-starter</artifactId>

<version>xxxxxx</version>

</dependency>

* 1. **排除不使用的pom依赖**
     1. 某个项目中需要使用到微信支付，支付宝支付，那么对应的pom应该如下配置(例举)

<dependency>

<groupId>cn.minsin</groupId>

<artifactId>mutils-spring-boot-starter</artifactId>

<version>版本号</version>

<exclusions>

<exclusion>

<groupId>cn.minsin</groupId>

<artifactId>mutils-dianwoda</artifactId>

</exclusion>

<exclusion>

<groupId>cn.minsin</groupId>

<artifactId>mutils-kuaidi100</artifactId>

</exclusion>

<exclusion>

<groupId>cn.minsin</groupId>

<artifactId>mutils-yiketong</artifactId>

</exclusion>

<exclusion>

<groupId>cn.minsin</groupId>

<artifactId>mutils-wechat-miniprogram</artifactId>

</exclusion>

<exclusion>

<groupId>cn.minsin</groupId>

<artifactId>mutils-file</artifactId>

</exclusion>

</exclusions>

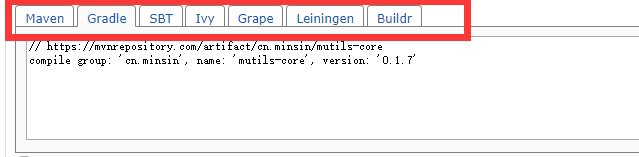
</dependency>

**说明:**

1.maven不支持想要某个依赖而添加哪个依赖，只能使用排除法。

**2.Mutils-core和mutils-spring-boot-autoconfigure不能被移除掉，因为它是整个框架的核心包。**

**3.**除maven外，同时也支持gradle、sbt等项目管理工具



**4.Mutils-core可以被项目单独引用，提供项目常用的帮助类如dateutil、stringutil、numberutil，ziputil等。附录1中将介绍mutils-core中的各类方法，依赖如下:**

<!-- https://mvnrepository.com/artifact/cn.minsin/mutils-core -->

<dependency>

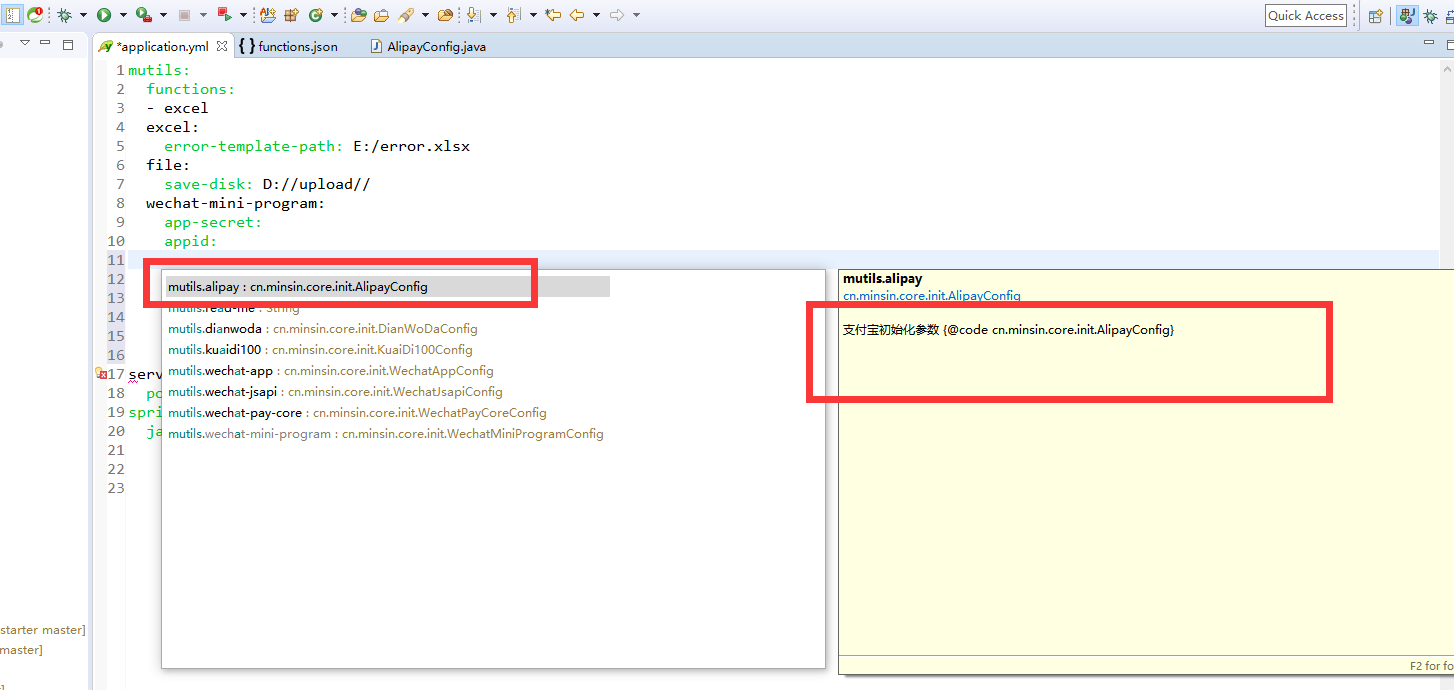
<groupId>cn.minsin</groupId>

<artifactId>mutils-core</artifactId>

<version>xxxx</version>

</dependency>

1. **开始上手**
   1. 在yml或properties中配置引用模块的参数(推荐使用yml)



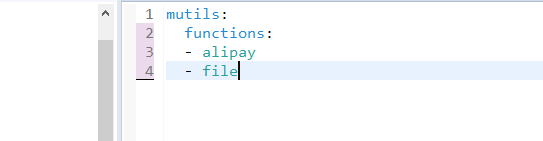


说明:

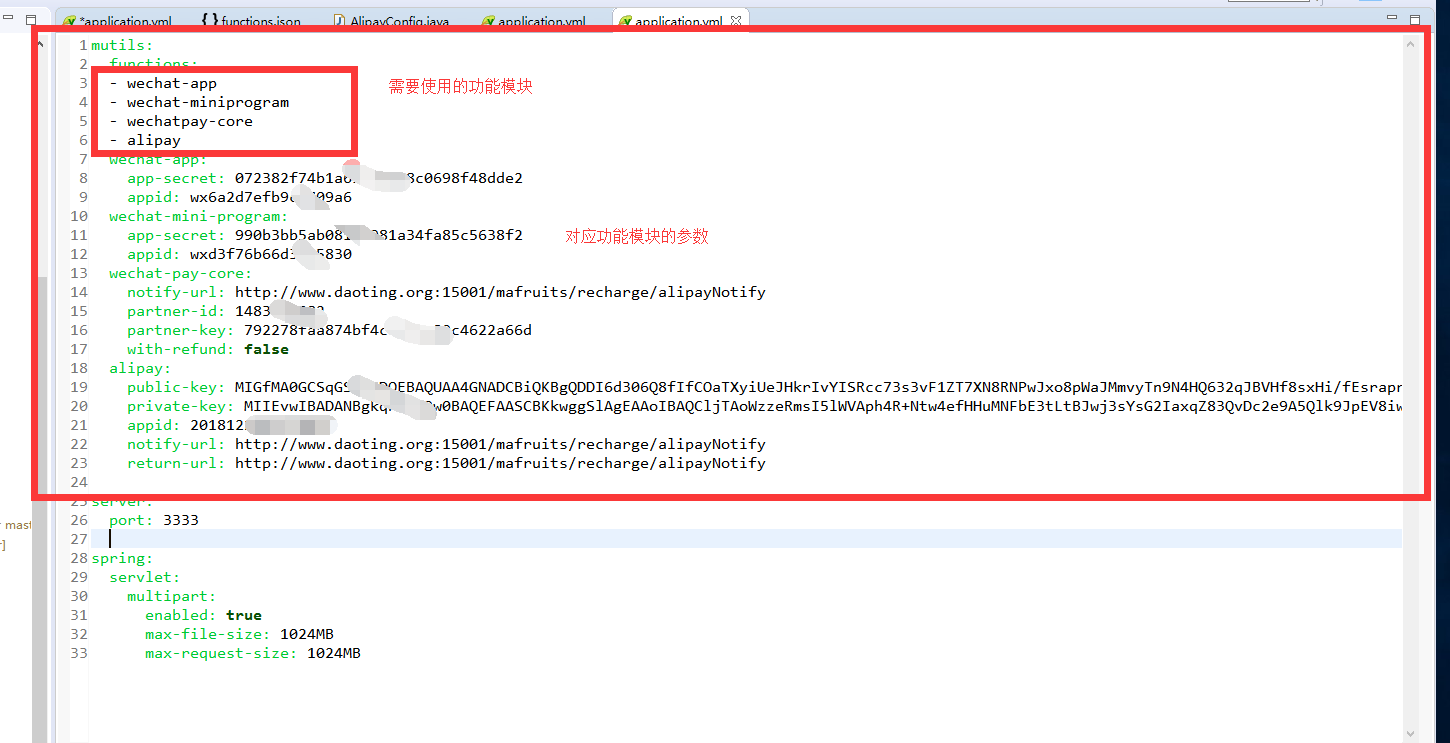
(1).在yml或者properties中输入**mutils**再按alt+/即可出现所有功能模块的配置参数。

其中每个模块都可以通过功能名字找到对应的config类，需要填写的参数都写在里面,可以找到对应的config去查看需要填写的参数，再根据实际情况进行填写。

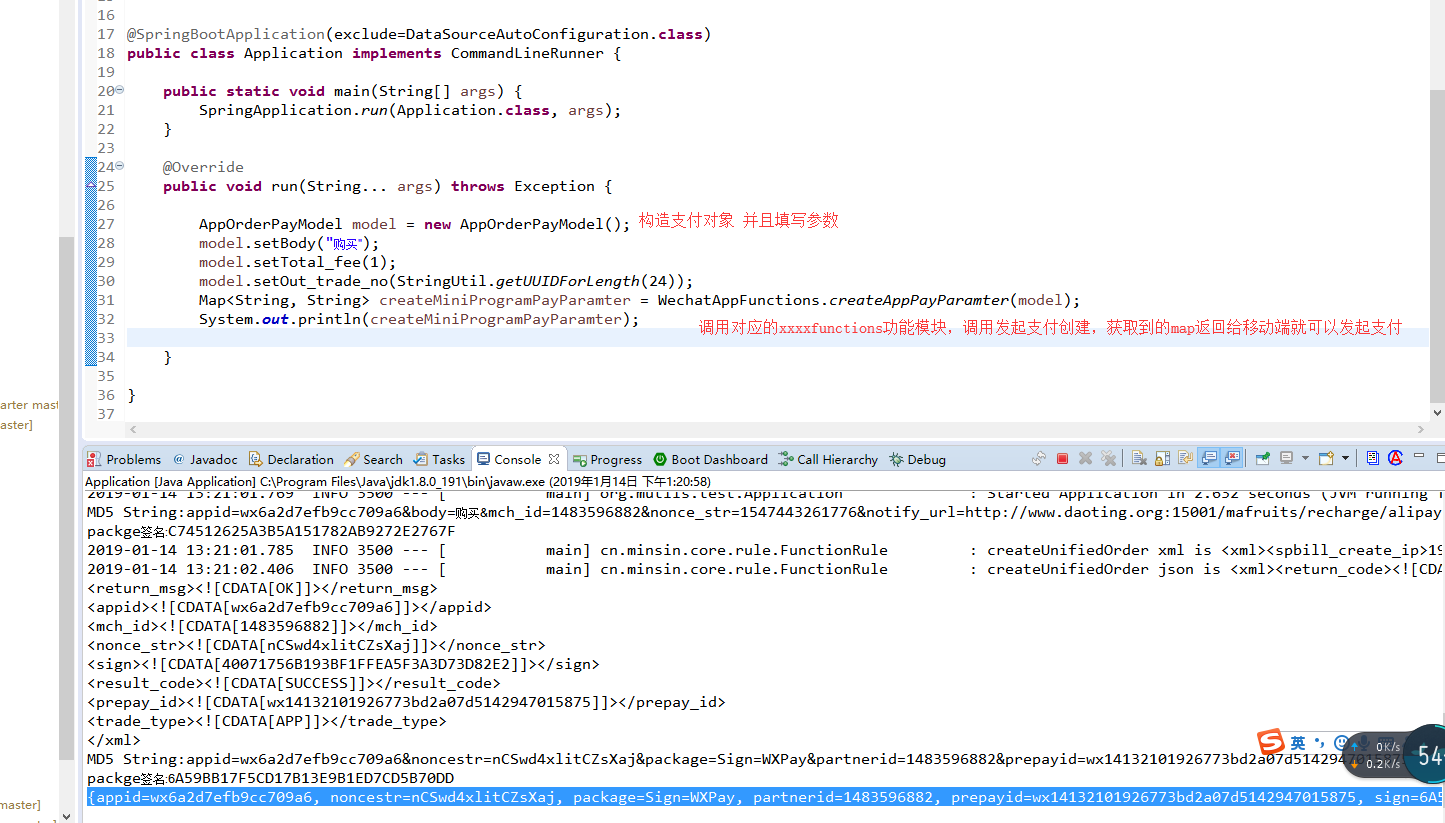
1. **在配置文件中，有一个属性叫做functions,它的配置将意味着项目启动时将会初始化哪些功能模块，只有在functions中填写的功能模块才会初始化并且进行检验参数合法性，如果没有填写functions 即使填写了对应的参数，也不会初始化的。以下为正确的填写例子:**



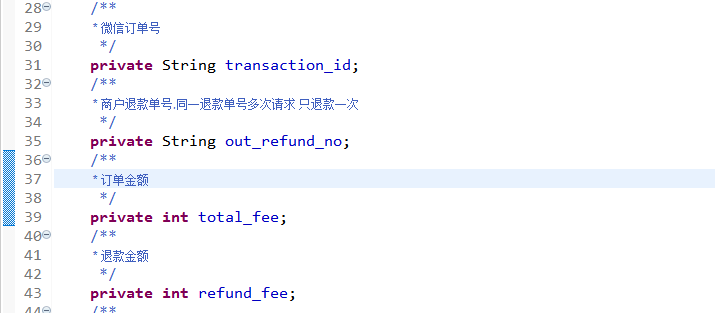
1. **使用微信支付、支付宝、点我达等业务功能模块**
   * 1. **业务形式的功能模块，都是需要再配置yml中填写对应的参数，这些参数将会在使用对应的功能模块自动配置，不需要手动赋值，因为项目启动时已经初始化好了。使用效果如下：**

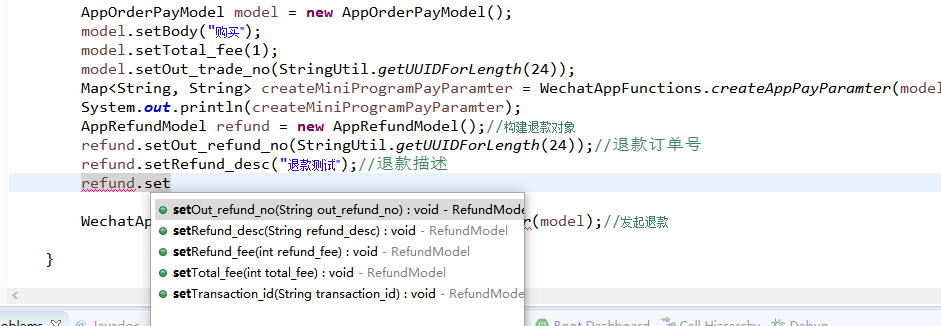


0.2.0版本中，将微信小程序、app、jsapi中的支付模块移动到了对应的功能模块中，如下将测试发起微信支付-app:



退款也是一样的：

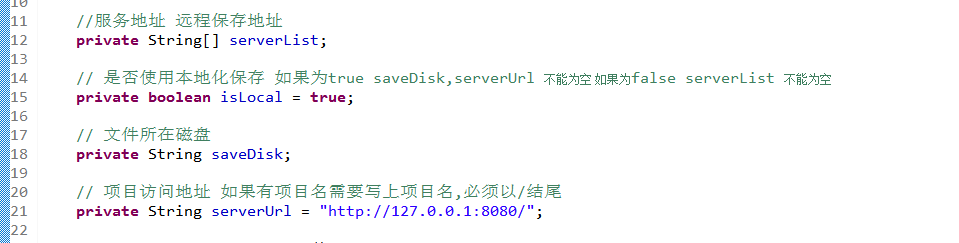




这个地方不能配置回调地址，回调地址统一在yml为notify\_url。签名是不能set的，签名由框架自动生成。

说明：所有功能模块的命名都是采用**xxxxFunctions**，有利于定位功能模块。提供的方法都是静态的方法可以通过类名直接调用，只需要传入对应的参数。

1. **使用file 在0.2.0版本中提供**
   1. **file是一个比较特殊的一个模块。用于统一规范文件上传和文件映射，举例如下：**



文件上传提供了两种方式进行存储文件，默认本地存储。

本地储存: isLocal为true 此时saveDisk必须填写并且要以/结尾,serverUrl也是必填的

远程储存: isLocal为false 时，serverlist不能为空，并且对应的地址要运行 mutils\_file\_server .

github地址：<https://github.com/mintonzhang/mutils-file-server>

然后在配置文件中，serverList填入mutils\_file\_server所在服务器，必须要以/结尾 如图所示：



**说明：对于本地存储**

**1.**如图所示，当调用FileFunctions中的saveFile时只需要把当时接收的文件传入，默认将会存入D:/upload/这个文件夹中。如果文件名重复，将以xxxx-副本的形式存在，如果副本也存在了将以xxxx-副本(i) i为从1到正无穷的数字进行递增。

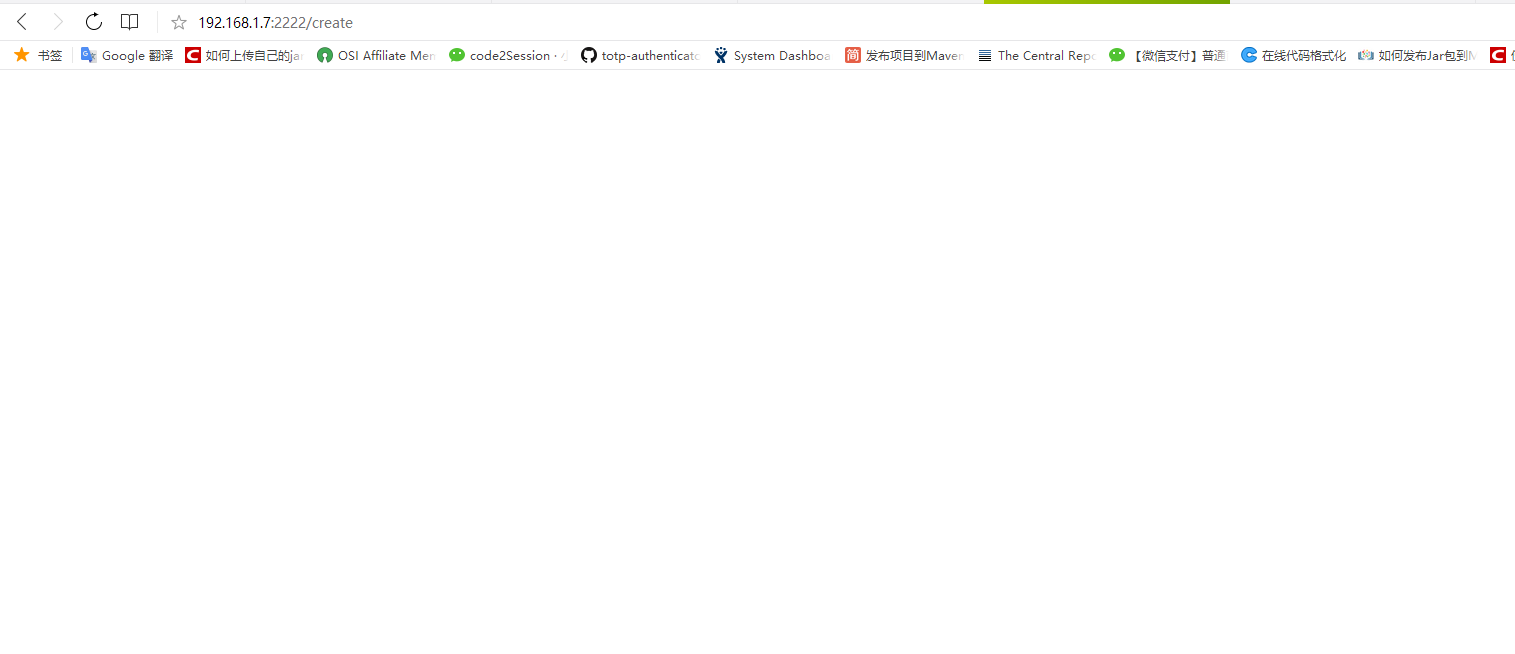
* 1. 如果启用了file这个模块，将会自动将save-disk的文件映射为静态资源，其访问路径为 域名/ip:端口/项目名/files/文件路径。 其中files是固定的，不能进行修改。
  2. Save-disk必须以/结尾，不然将会初始化失败。
  3. 在配置文件中填写serverurl 即填写项目的初始访问路径 不需要具体到功能

例如: <http://192.168.1.7:8080/pro_dianxin/>

这样的地址 必须使用/结尾

当我们要获取项目时，使用FileFunctions.getRequestUrl 来获取文件的网页访问路径。 注意:此功能在0.1.9中才会有

1. **使用excel**
   1. Excel在项目中是比较频繁使用的一个模块。因为我们提供的文件下载，不会使用ajax进行下载。此时如果服务端出现问题，展现给用户的提示是并不友好的。使用前：

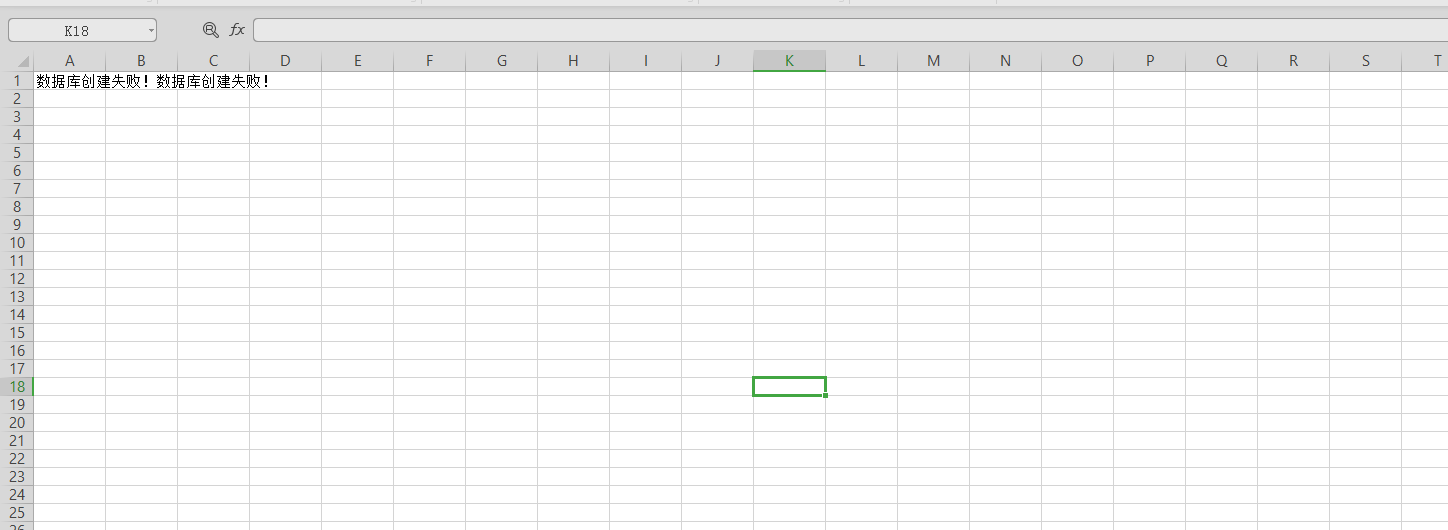




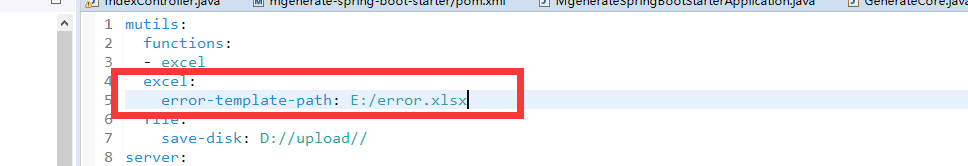
**这样的提示给用户，将会是一个A类bug.**

使用后：





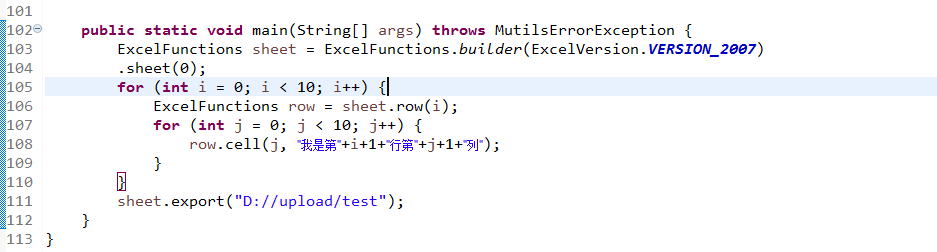
此外，这样的错误模板是可以自定义的。在yml中的配置如下：



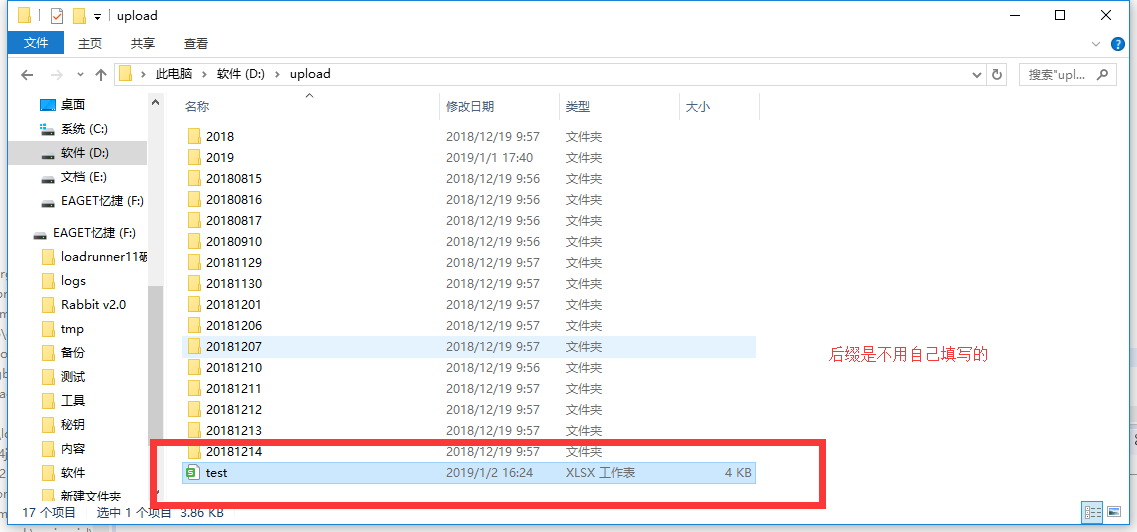
我们只需要指定一个错误的模板位置，当导出时出现错误，开发者手动捕捉，调用ExcelFunctions中的error方法，将会输出此文件作为导出内容。**在0.1.9+，导出功能出现错误会默认将错误模板当做本次下载的输出对象，从而避免向页面上抛出500或404的错误。**

当然，excelFunctions中不会仅仅只提供一个功能给开发者，方法介绍如下：

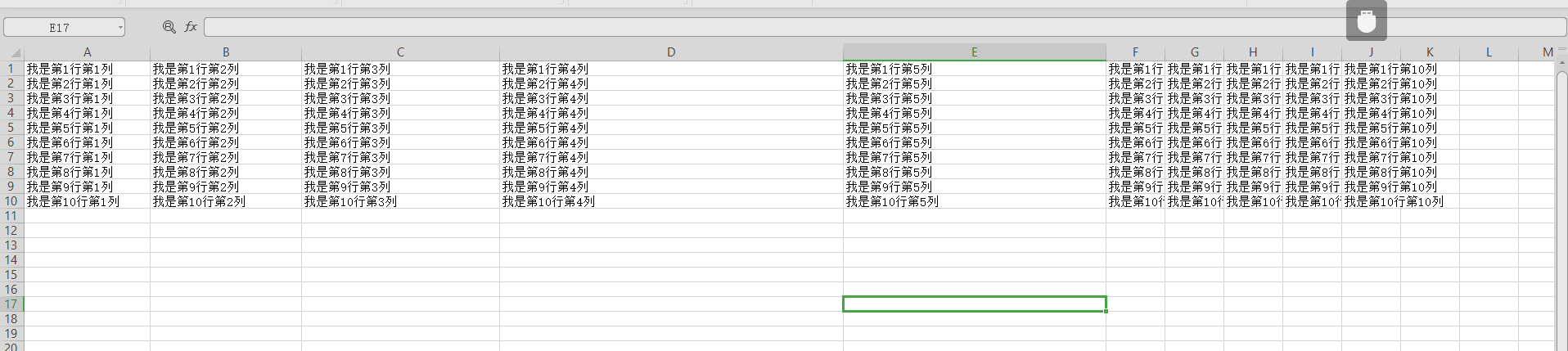
1. 解析excel(解决兼容性问题，无论是2007还是2003，都能解析)
   1. 导入作为第一次使用excel的开发人员来说，都是一个很费神的事情。现在只需要使用ExcelFunctions.bulider去加载excel，再使用getCellValueList 传入要解析的sheet和开始行数及准备解析的列就可以获取到指定的所有数据
2. 导出excel
   1. 采用构建者模式的导出极为简单和易读，举例如下：



简单的代码实现导出10行10列的excel

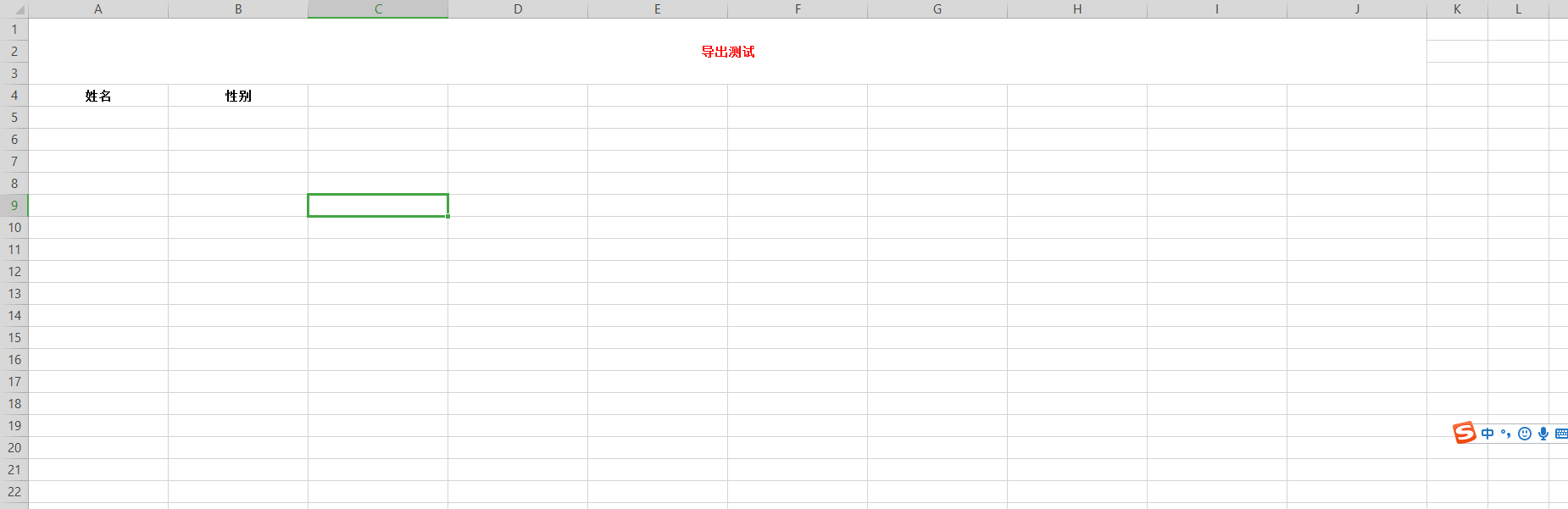


**导出文件的文件后缀是不用手动填写的，会自动根据bulider时所选的version进行创建。**

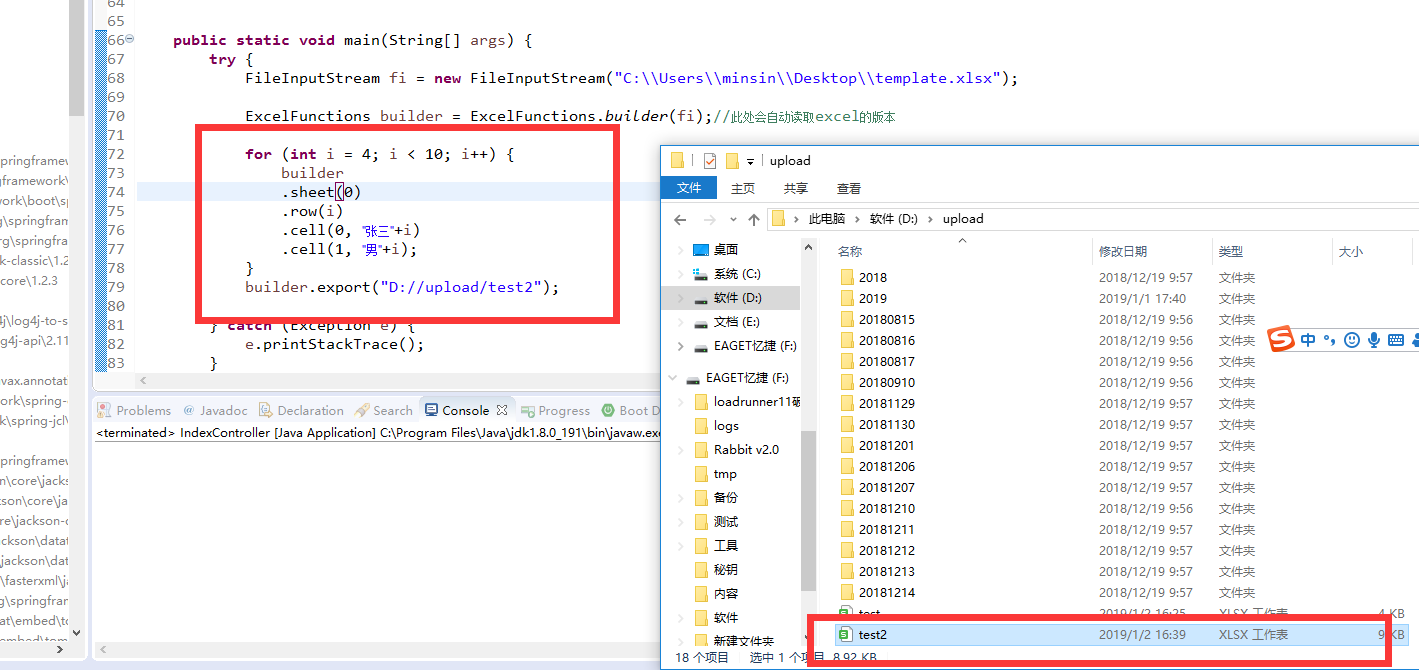


这样就导出了一个10行10列的excel，不仅如此，我们可以自定义模板文件，在builder的时候(ExcelFunctions.builder(InputStream in))，加载进去，在进行填充数据，导出时将会按照原有的格式导出。代码如下:

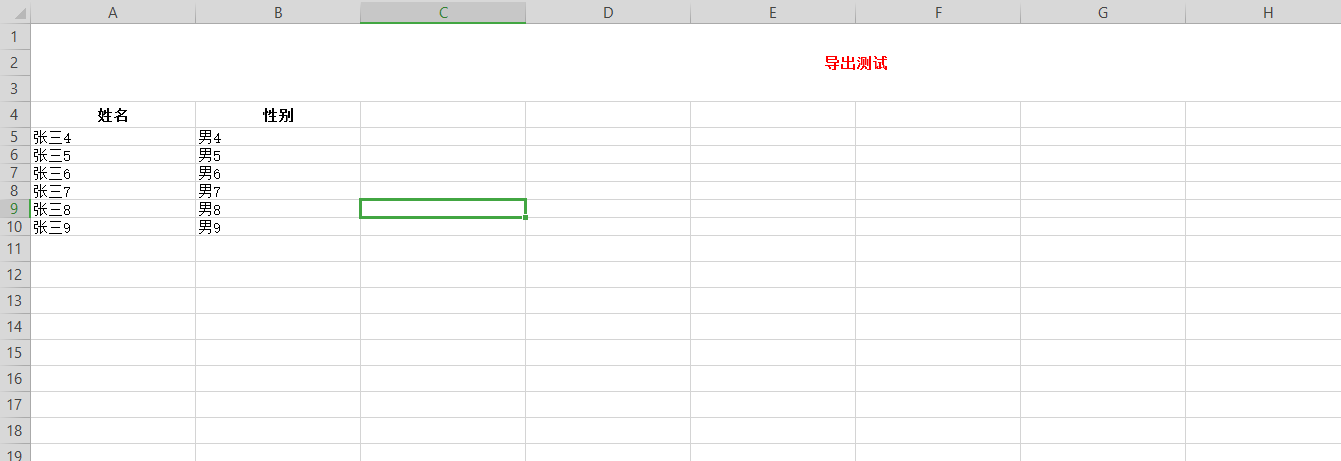
1. 准备一个模板



1. 代码和导出文件



1. 结果



Export有两个重载

**例子中是输出到本地文件，它的另一个重载是输出到浏览器，让用户点击下载。需要多传入**HttpServletResponse对象

3.加载excel  **builder**

4.解析excel的真实值 **getCellRealValue**

5.加载resource下的模板文件  **getExcelTempalte** 只能加载resource下面的文件

6.错误文件导出 直接调用该方法将会读取配置文件中的错误模板作为输出对象

附录1：

模块名：Mutils-core：

包名：

1. cn.minsin.core.exception
   1. 框架内自定义异常类。在框架开发中将会抛出该异常
2. cn.minsin.core.rule
   1. 其中functionRule和modelRule为拓展第三方API中所建立的规范父类。其中不能定义属性。所有的APIModel和Functions都需要集成对应的父类。
3. cn.minsin.core.tools
   1. Stringutil 提供单个字符串或多个字符串的非空验证，生成UUID，全文去空格，收尾去空格的多种方法。
   2. DateUtil 提供时间转字符串，long转字符串，字符串转时间，时间大小比较等多种常用时间判断
   3. NumberUtil 提供数值之间的互转(toLong 、toInt)，判断是否为指定类型等常用方法
   4. MapUtils 提供对象转Map 及动态类型创建map
   5. ListUtil 提供动态类型创建list，判断list是否为空等常用list方法
4. cn.minsin.core.web

Web 中提供了服务端返回给前端的消息定义及结构定义的相关实体类

* 1. VO 的出现替换了之前频繁创建VO实体类，只需要一个VO，简化了BO和VO之间的转换，并且提供**getObject() 方法转换为指定对象**
  2. page中提供页码检查，条数检查，及物理分页的相关方法
  3. Result 作为服务端返回给前端的重要返回对象。为了避免code中使用1 2 3 这种返回值作为code. 采用枚举类型定义code码和返回消息。 在0.1.9中定义ResultOptions接口进行统一规范。默认使用DefaultResultOptions

通常情况下 DefaultResultOptions中提供的code和msg是够用的，如果有特殊案例，需要实现ResultOptions 进行枚举拓展

使用Result.buliderSuccess()等方法进行创建不同对象