# NOVEL SPIDE

1. 简单说明：

目前改工程分为两大模块：搜索引擎、站点解析。

* 搜索引擎：

通过百度、搜狗、神马类似的搜索引擎来识别小说，并解析。也可针对站内搜索引擎来进行爬取，如：起点、17K、笔趣阁等。

通过搜索引擎我们可以获取到小说的基本信息。如：书名、作者、简介等，但是获取不到全部信息，比如：章节，这时就需要站点解析。

* 站点解析

站点解析是通过该小说的地址来获取小说的信息，这时通过站点解析模块，可以获取到站点下该小说的所有信息。注意这是是通过小说地址来获取小说信息，并不是书名，如果通过书名获取小说信息，那么就需要用到搜索引擎。

**总结：**

搜索引擎的存在是用来查找小说，只能获取小说的基本信息，而站点解析是针对该站点获取到的小说信息。

1. 依赖环境

NovelSpider依赖的框架：WebMagic(主)和guava(副)。

* WebMagic

webmagic是一个开源的Java垂直爬虫框架，目标是简化爬虫的开发流程，让开发者专注于逻辑功能的开发

* Guava

Guava是使用webmagic框架的Scheduler组件的一个依赖包。

具体说明：<http://webmagic.io/docs/zh/posts/ch6-custom-componenet/scheduler.html>

* 编译环境

操作系统：Win10

开发工具：IDEA 2018.1.5

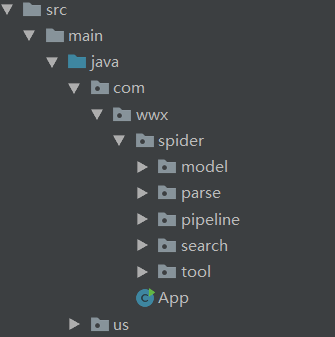
仓库管理: maven

JDK版本：1.8

# 快速开发

一：项目结构

如右下图：

 该工程共4个大包(us包是修复了webmagic框架不能访问HTTPS TL3协议的问题,可忽略,pipeline包已经作废)，

* Model：该包名是用来存放Bean
* Parse: 该包就是两大模块之一的站点解析包
* Search: 该包就是两大模块之一的搜索引擎包
* Tool : 工具包

二：加入开发

在开始之前强烈建议你看webmagic框架的一个小结：

<http://webmagic.io/docs/zh/posts/ch2-install/first-project.html>

这是一个极其简单的入门实列，如果你能看完整个文档(不需要全部知道怎么使用)，那就更好了，你需要重点看Page和Site类的相关操作，这在我们下面开发很重要。

# 创建搜索引擎：

如果你觉得默认的搜索引擎不给力，可以自写一个搜索引擎，其实开发一个搜索引擎很简单，这里就拿默认的神马搜索引擎来举例了：

首先你需要在Search包中创建一个搜索类，并继承AbSearch类：

public class SmBookSearch extends AbSearch

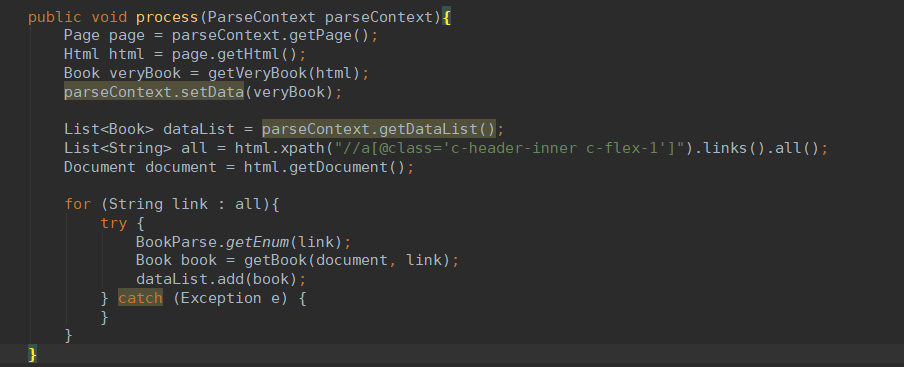
对就像上面这样，然后你需要实现2个方法getUrl(String name)和process(ParseContext parseContext)。

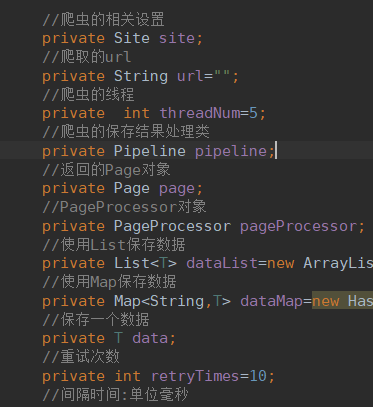
getUrl：你要告诉爬虫，需要去爬哪一个地址,那个name参数就是书名啦。



Process：这是一个解析方法，你应该在这里面注重于如果解析到想要的数据。parseContext是一个解析上下文对象，这个下面会说。

目前可以透露一点的是parseContext. getPage()方法是可以返回一个Page对象，然后在调用getHtml()，就可以获取到HTML对象，具体HTML怎么操作，可以看这个链接：<http://webmagic.io/docs/zh/posts/ch4-basic-page-processor/selectable.html>





parseContext类的常用API：

dataList：将数据以List的形式存放

dataMap: 将数据以Map的形式存放

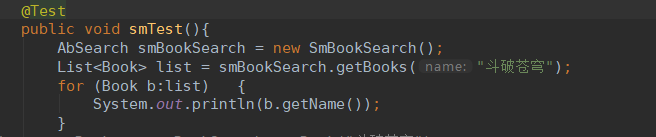
data:存放一个数据

site：获取到Site对象，这是webmagic框架里面的

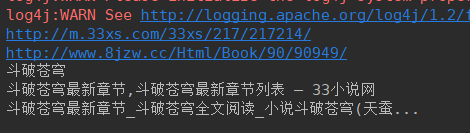
更多操作，可以查看源码，都注释啦。

注意：解析好的Book类一定要添加到parseContext类中，如果是一个数据就存放在data中，如果是多个就存放在dataList中，Map数据可以自由发挥。

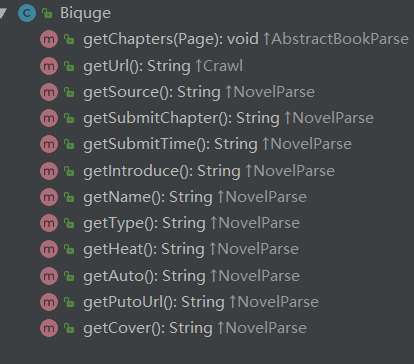
哈哈，完成2个方法，那么搜索引擎就已经完成了，看看我们如果调用吧



运行结果：



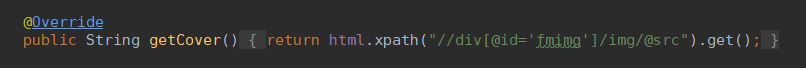
# 创建站点解析

搜索引擎可以少，但是站点解析少了，那么工程就不强大了，如何创建一个解析呢？也很简单，只需要继承AbstractBookParse类可以了，那么我们用笔趣阁，来做个解析栗子吧。

首先继承AbstractBookParse类，这个时候你需要实现一下方法：

别慌，看到这么多方法，不要腿软了，其实很简单。这些都是Book类的get属性，不信你去看看。

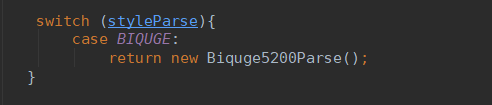
继承了AbstractBookParse类以后呢，你会有一个html类，这是webmagic框架里面的，基本一个get方法里面就一行代码：

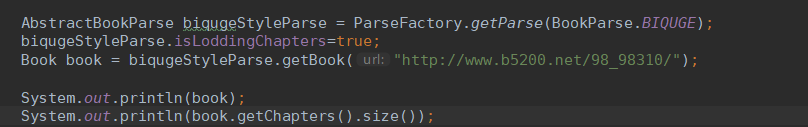


然后你也可以直接在方法里调用parseContext类。

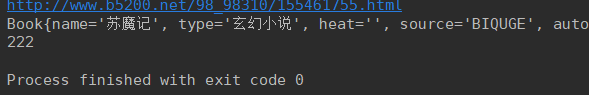
注意: getChapters 方法需要把章节存放parseContext类的datalist中去。(具体可以看实咧源码)其他直接返回就好。

写完后，不要急着运行。请在BookParse里创建你写好的站点枚举类。

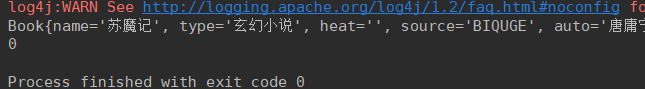
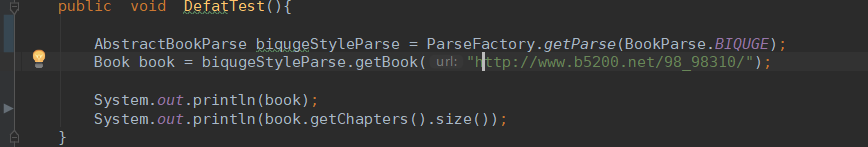
然后在ParseFactory简单工厂类中填加上你实现的解析类，这样就真正完成了一个站点解析的实现。

来吧，看下效果：

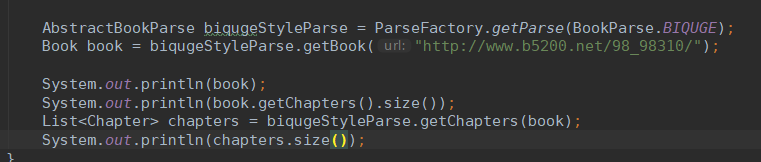
运行结果：



isLoddingChapters：是指在创建时是否把章节一起加载过来，默认为false；



需要单独获取章节就这样：

当然也可以通过url，来获取章节。其效果是一样的。