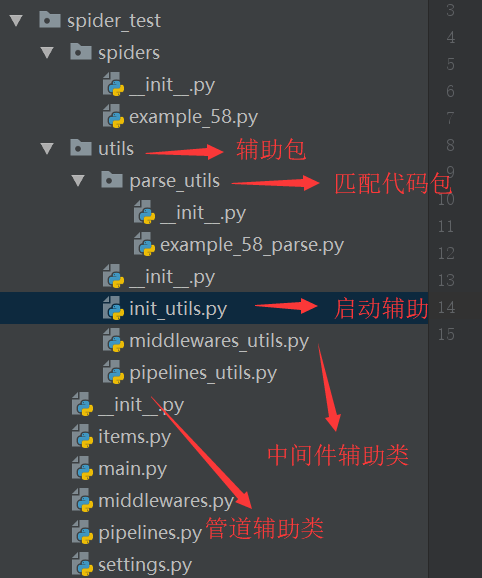
1. 开发目录规范统一。
2. 匹配代码与抓取分离，将匹配代码统一写在parse\_utils里面。
3. 代理池，cookies池，反爬特征，尽量在下载中间件中解决。
4. 辅助方法统一规范。



1. 根据爬取周期确定使用架构。
2. 抓取周期在一个小时内，不需要对已经抓取的链接做持久化。
3. 抓取周期在一天内（或抓取的数据总量30w以下），需要对抓取链接做一个相对简单的持久化。
4. 抓取周期在一天以上，需要相对专业性的链接持久化。

对于已经抓取过的链接，进行持久化，并且在启动的时候加载进入去重队列，是一个比较强的需求，主要应对爬虫故障重跑，不需要重跑所有链接。

目前已经寻找到三种方案解决。

需要大家根据情况进行处理。

1. 已经造好的轮子系列

scrapy-deltafetch, scrapy-crawl-once, scrapy-redis, scrapy-redis-bloomfilter

B．目前还有一个我写工具在init\_utils方法里面，可以轻量实现。

1. 爬虫的文档，重构，和方法规范。
2. 邮件报警，机制分级。
3. 项目中包含 data 文件夹存储数据，包含清洗数据程序的文件夹, 包含运行脚本
4. 尽量采用广度写爬取逻辑，递归，

爬虫请求应该包含 构建请求函数build\_xxx\_request、

解析请求函数 parse\_xxxx\_response

1. 编写逻辑文档，或者思维导图，说明流程。