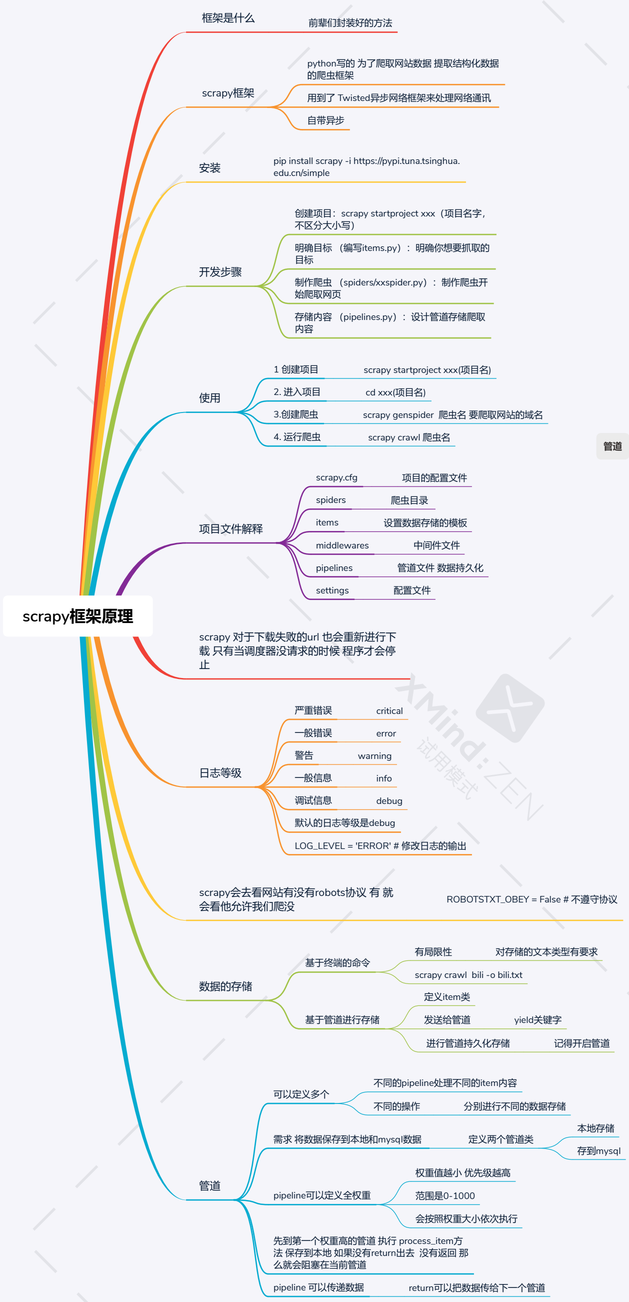
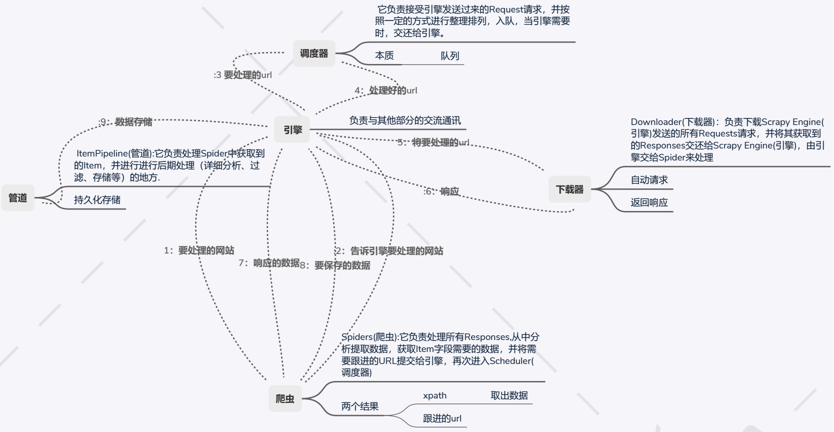


scrapy框架原理

- 框架是什么
  - 前辈们封装好的方法
- scrapy框架
  - python写的 为了爬取网站数据 提取结构化数据的爬虫框架
  - 用到了 Twisted异步网络框架来处理网络通讯
  - 自带异步
- 安装
  - pip install scrapy -i https://pypi.tuna.tsinghua.edu.cn/simple
- 开发步骤
  - 创建项目：scrapy startproject xxx（项目名字，不区分大小写）
  - 明确目标 （编写items.py）：明确你想要抓取的目标
  - 制作爬虫 （spiders/xxspider.py）：制作爬虫开始爬取网页
  - 存储内容 （pipelines.py）：设计管道存储爬取内容
- 使用
  - 1 创建项目
    - scrapy startproject xxx(项目名)
  - 2. 进入项目
    - cd xxx(项目名)
  - 3.创建爬虫
    - scrapy genspider 爬虫名 要爬取网站的域名
  - 4. 运行爬虫
    - scrapy crawl 爬虫名
- 项目文件解释
  - scrapy.cfg
    - 项目的配置文件
  - spiders
    - 爬虫目录
  - items
    - 设置数据存储的模板
  - middlewares
    - 中间件文件
  - pipelines
    - 管道文件 数据持久化
  - settings
    - 配置文件
- scrapy 对于下载失败的url 也会重新进行下载 只有当调度器没请求的时候 程序才会停止
- 日志等级
  - 严重错误
    - critical
  - 一般错误
    - error
  - 警告
    - warning
  - 一般信息
    - info
  - 调试信息
    - debug
  - 默认的日志等级是debug
  - LOG_LEVEL = 'ERROR' # 修改日志的输出
- scrapy会去看网站有没有robots协议 有 就会看他允许我们爬没
  - ROBOTSTXT_OBEY = False # 不遵守协议
- 数据的存储
  - 基于终端的命令
    - 有局限性
      - 对存储的文本类型有要求
    - scrapy crawl bili -o bili.txt
  - 基于管道进行存储
    - 定义item类
    - 发送给管道
      - yield关键字
    - 进行管道持久化存储
      - 记得开启管道
- 管道
  - 可以定义多个
    - 不同的pipeline处理不同的item内容
    - 不同的操作
      - 分别进行不同的数据存储
  - 需求 将数据保存到本地和mysql数据
    - 定义两个管道类
      - 本地存储
      - 存到mysql
  - pipeline可以定义全权重
    - 权重值越小 优先级越高
    - 范围是0-1000
    - 会按照权重大小依次执行
  - 先到第一个权重高的管道 执行 process_item方法 保存到本地 如果没有return出去 没有返回 那么就会阻塞在当前管道
  - pipeline 可以传递数据
    - return可以把数据传给下一个管道



- **调度器**
  - 它负责接受引擎发送过来的Request请求，并按照一定的方式进行整理排列，入队，当引擎需要时，交还给引擎。
  - 本质
    - 队列
- **引擎**
  - 负责与其他部分的交流通讯
- **下载器**
  - Downloader(下载器)：负责下载Scrapy Engine(引擎)发送的所有Requests请求，并将其获取到的Responses交还给Scrapy Engine(引擎)，由引擎交给Spider来处理
  - 自动请求
  - 返回响应
- **管道**
  - ItemPipeline(管道):它负责处理Spider中获取到的Item，并进行进行后期处理（详细分析、过滤、存储等）的地方.
  - 持久化存储
- **爬虫**
  - Spiders(爬虫):它负责处理所有Responses,从中分析提取数据，获取Item字段需要的数据，并将需要跟进的URL提交给引擎，再次进入Scheduler(调度器)
  - 两个结果
    - xpath
      - 取出数据
    - 跟进的url

1：**要处理的网站**

2：**告诉引擎要处理的网站**

:3 **要处理的**url

4：**处理好的**url

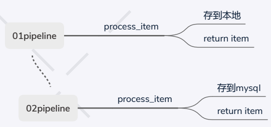
5：**将要处理的**url

:6：**响应**

7：**响应的数据**

8：**要保存的数据**

:9：**数据存储**



- 01pipeline
  - process_item
    - 存到本地
    - return item
- 02pipeline
  - process_item
    - 存到mysql
    - return item

XMind: ZEN | 试用模式