CS209 Final Project

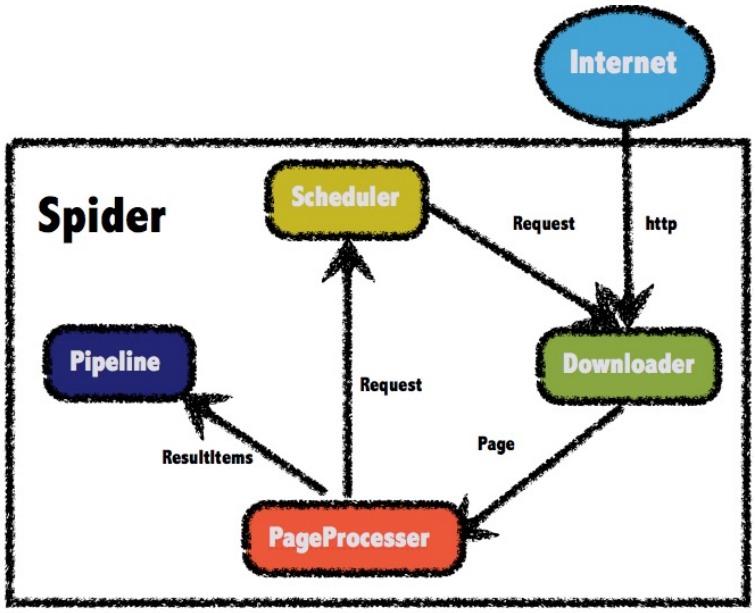
> Member: 黄明朗(12011828) 罗俊杰(12011112) 黄杭(12010915)

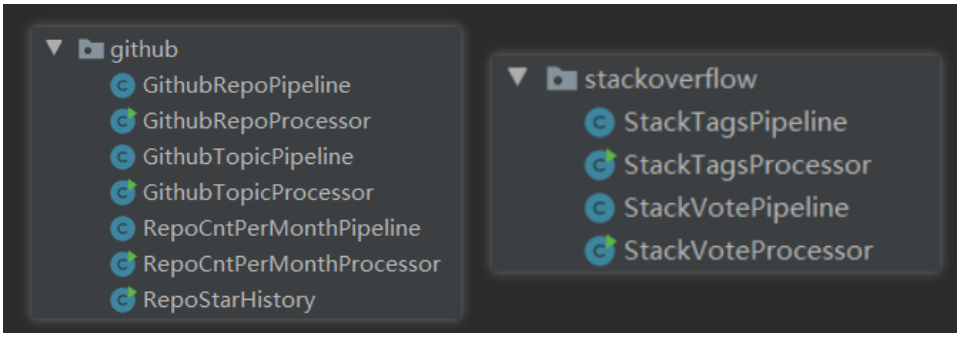
总体介绍

Spring boot + Mybatis-plus + MySQL + LayUI + Eccharts

使用工具：IntelliJIDEA Ultimate、Datagrip

使用WebMagic爬取网站数据





▼ 使用Restful API爬取github\stackoverflow 对应的内容

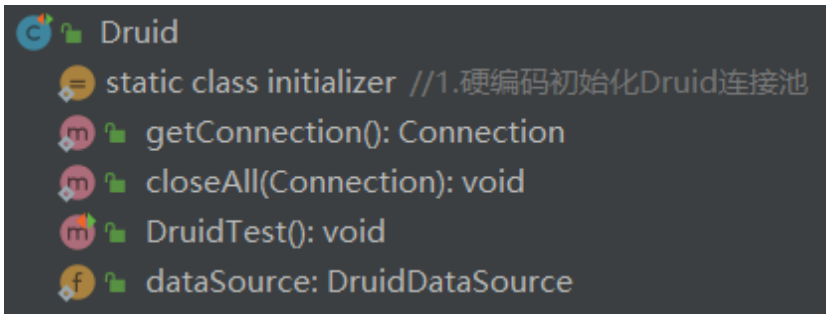
- Spider 通过`addUrl` 将网页转为Page类 输送给PageProcessor, PageProcessor 通过 getRawText方法获取网页文字内容
- 根据webmagic 爬取到的 相关link, 我们使用HttpClient、HttpGet进行二次爬取以获取有用信息
- 使用Webmagic 提供的`JsonPathSelector` 对爬取的网页的Json字符串内容进行解析，并将解析到的内容 转化为Entity 中对应的实体对象 存到List中，通过page.putField()方法传送到对应Pipeline类中

▼ 多线程爬取数据

- `Processor`利用webmagic 自带的Spider.thread() 方法实现多线程爬取网页内容
- `Pipeline` 利用 ReentrantLock 实现多线程+批处理 进行数据导入

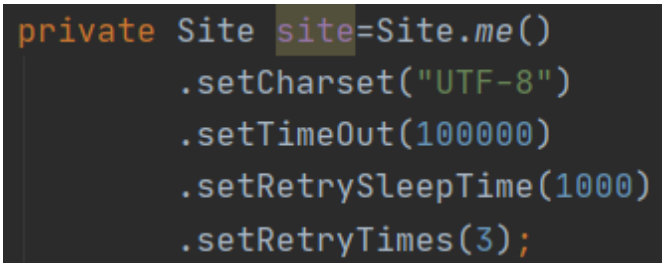
```
public Lock balanceChangeLock =new ReentrantLock();
balanceChangeLock.lock();
//...
    try{
        openDB("webmagic");
        //...导入数据进数据库
        stmt.executeBatch();
        commit();
        closeDB();
    }finally {
        balanceChangeLock.unlock();
    }
```

▼ 使用`JDBC`、`Druid数据库连接池` 、`数据库批处理机制`将爬取的数据导入到数据库中



▼ 利用`webmagic`、`HttpClient`、`Jsoup`等自定义设置数据爬取方式

- 利用webmagic 自带的方法实现如字符集\爬取时间\尝试次数等设置，避免因为爬取速度过快而IP被封，影响爬取效率



```
private Site site=Site.me()
        .setCharset("UTF-8")
        .setTimeOut(100000)
        .setRetrySleepTime(1000)
        .setRetryTimes(3);
```

- 利用HttpGet、HttpClient设置header、accessToken等，方便爬取

```
HttpGet httpGet = new HttpGet(url);
httpGet.setHeader("Accept","application/vnd.github.v3.star+json");
httpGet.setHeader("Authorization","token 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");
```

后端

- 数据持久层框架：`Mybatis-plus`
- WEB 框架：`Spring MVC`
- `Shiro` 实现网站用户管理

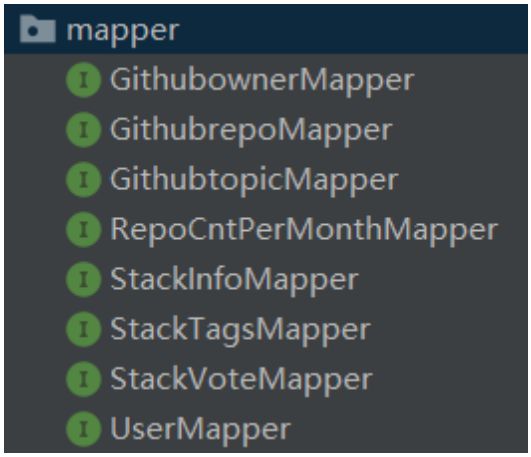
▼ Entity实体类 (POJO)

我们使用Lombok为实体类配置 setter，getter，equal等方法, 减少代码量

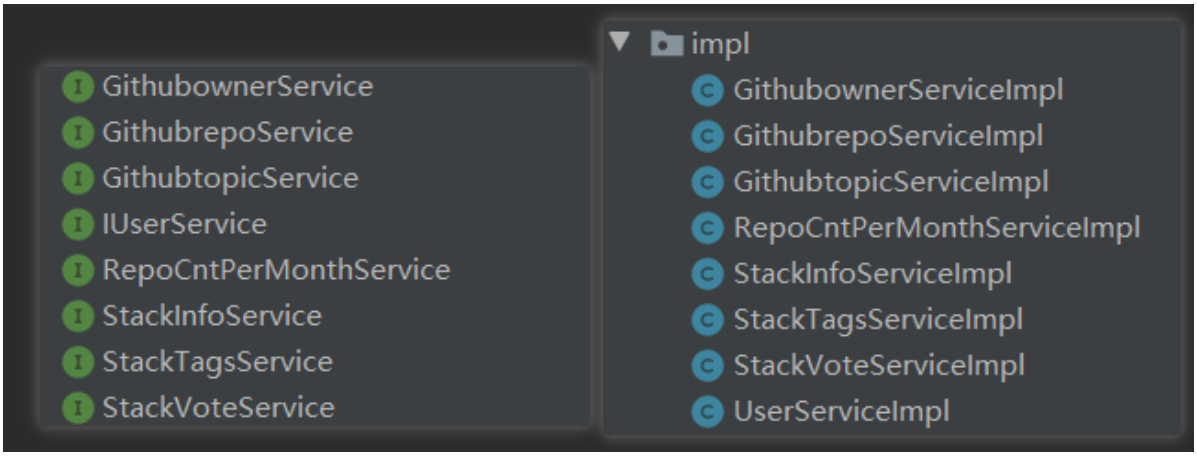
- `GithubOwner` ——— Github 上的 优秀repo作者
- `GithubRepo` ——— Github 上的优秀repo
- `GithubTopic` ——— Github 上优秀repo对应的topics集合
- `RepoCntPerMonth` ——— Github 每年每月repo 的创造数据集合
- `StackInfo` ——— StackOverflow 对于score 的集合，分为四类(1. questioner, answer, 最多被view的问题，最多被answer的问题集合)对应的score集合
- `StackTags` ——— StackOverflow上的优秀问题对应的tags集合
- `StackVote` ——— StackOverflow上的优秀问题相关信息的集合

- `User` ——— 对应网站所登记的用户集合

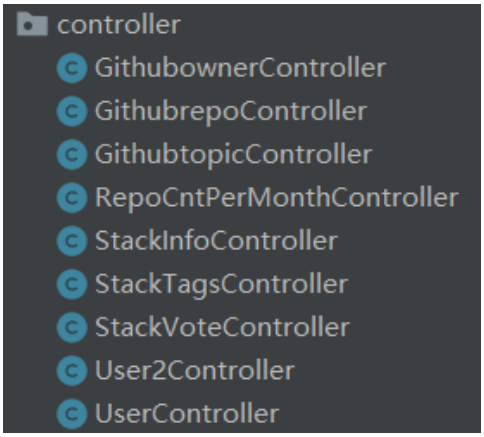
▼ Mapper 类 (DAO)



▼ Service 类 (分为Service接口 以及ServiceImpl 接口实现类)

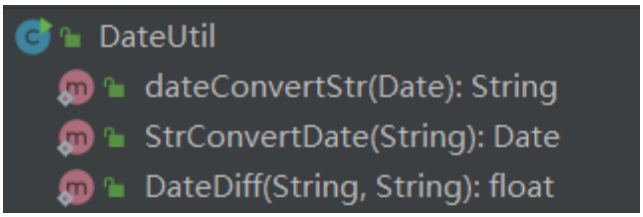


▼ Controller类

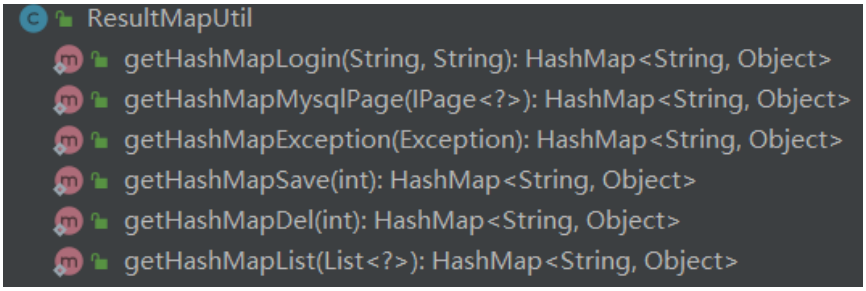


▼ Util 类

- DateUtil
  处理Java.Util.Date 与String 之间的相互转换，以及Date 数据类型的比较



- ResultMapUtil
  用于整合并返回 用户在网页上所请求行为和信息 对应的后台反馈，适配LayUI 前端框架
  如用户进行Login 操作时，`getHashMapLogin` 则根据后端结果会返回hashmap，提示用户是否登录成功。



▼ 配置类

- `Shiro` 实现网站用户信息管理

- Mybatis-plus `@Configuration` 注解实现对项目的配置

前端

▼ 使用Thymeleaf 模板引擎 动静分离渲染网页内容

▼ 前端UI框架

- 使用`Layui`搭建前端主题框架
- 使用`Echarts` 实现网页数据可视化
- 相关前端效果 来源于https://echarts.apache.org/zh/index.html

▼ 前端JS 框架

- JQuery
- `ajax`局部网页动态渲染

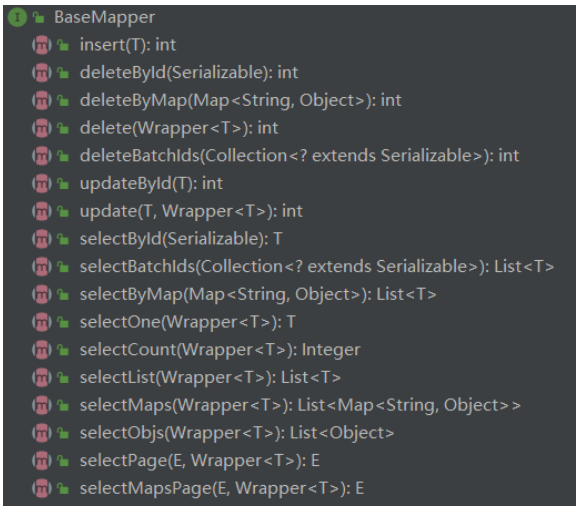
▼ 网页设计结构

- 使用LayUI 导航组件，将不同主题的内容分离开来，增强可观性
- 利用LayUI提供的数据表格组件，将获取到的数据以表格的形式展现出来
- 利用LayUI的按钮组件，与后端数据库连接，实现在前端 对 某些数据的增删查改操作

数据分析

▼ 数据分析过程提取

- 利用`Mybatis-plus`中的`BaseMappper<?>` 接口 提供的CRUD 方法 直接在java代码层面对数据 进行筛选，减少了对xml、jdbc等的编写压力



- 使用JDBC\SQL 聚合函数等 实现对原始数据的提炼

```
select startime2,repo,
sum(starnum) over (partition by repo order by startime2) as starsum
from repostarhistory
where repo in
(select t.name from (select * from githubrepo g order by g.star desc limit 10)as t)
order by startime2;
```

▼ 数据分析结果

- 我们的Topic涉及 Github中优秀Java Repo 的star、fork、月均star数量、Star History；Github 网站每年每月repo的创造数量，以及热门repo star热度变化；Github流行topic等等
- 涉及Stackoverflow中优秀的Java相关问题；优秀的Java问题回答者；Java问题中热门的topic等等

在进行对Github、StackOverflow上爬取的数据进行分析之后，我们发现：

- Github 中优秀的、受欢迎的Repo 作者有如Snailclimb、macrozheng、JakeWharton、MisterBooo等，而受欢迎的Repo Organization有apache、alibaba、google、Netflix、tencent等; 热门关键词有Android、spring-boot、hacktoberfest等
- 而StackOverflow中 优秀的Questioner 有GManNickG、user4315、Freewind、ashogelal等，优秀的Answerer有Mystical、BalusC、Jon Skeet等； 热门问题关键词有string、android、java-8、collections、spring等等