# 项目总结报告

豆瓣电影数据可视化

小组名称： 第五组

学员姓名： 黄学伟

目录

[项目总结报告 1](#_Toc17289904)

[1. 项目基本信息 2](#_Toc17289905)

[2. 项目成果 2](#_Toc17289906)

[3. 开发工具与环境 2](#_Toc17289907)

[4. 项目需求变更 3](#_Toc17289908)

[5. 项目计划和进度实施 3](#_Toc17289909)

[6. 经验总结 3](#_Toc17289910)

[7. 教训总结 4](#_Toc17289911)

[8. 其他改进意见 4](#_Toc17289912)

# 项目基本信息

该项目通过爬取豆瓣电影数据，根据具体需求进行

# 项目成果

主要负责：2017-2018两年电影数据的爬取



# 开发工具与环境

Node.js：前端应用开发方法

Cheerio：实现HTML数据解析

MongoDB：对需要的数据进行存储

eCharts：数据可视化展示

superagent：客户端请求代理模块（支持HTTPS）

# 项目需求变更

(主要说明在项目开发过程中项目出现的变更需求，出现变更的原因，以及变更导致的结果和影响等)

需要爬取的数据需求变更：

第一次：爬取名字、评分、国家、年份、评论人数

第二次： 爬取名字、评分、国家、年份、导演

最后一次： 爬取名字、评分、国家、年份

变更原因：电影子页面没有评论人数、图表中没有需要导演数据的内容

导致的结果和影响：修改爬虫代码cheerio部分

需要存储的数据需求变更：

第一次：存储所有电影

第二次：只存储评分、年份不为空的电影数据

变更原因：存储空评分或年份不利于数据统计

导致的结果和影响：修改爬虫代码cheerio部分及存储数据库部分

# 项目计划和进度实施

一共十年数据，每个人爬两年数据，剩余一人完成数据可视化部分内容

# 经验总结

1. 项目开始前的规划和对边界问题的考虑可以有效提供开发效率
2. 大部分网站都具有反爬虫机制，要想降低被封IP几率，最好使用代理进行爬取，或者使用同步操作进行模拟人手点击。

# 教训总结

编写代码的时候完全没有思路只能改改老师给的简单例子，因为没有设置代理，所以反爬虫机制让我很难受，需要加强学习。

# 其他改进意见

对项目的开发过程有其他可以优化和改进的地方