# 项目技术说明书

豆瓣电影数据可视化

小组名称： 第五组

学员姓名： 唐浩彬

目录

[项目技术说明书 1](#_Toc17289747)

[1. 项目介绍 2](#_Toc17289748)

[2. 需求分析 2](#_Toc17289749)

[3. 技术选型 3](#_Toc17289750)

[4. 爬虫开发和优化 3](#_Toc17289751)

[5. 数据清洗和存储 4](#_Toc17289752)

[6. 数据查询和可视化展示 5](#_Toc17289753)

[7. 开发环境 5](#_Toc17289754)

[8. 产品原型 6](#_Toc17289755)

# 项目介绍

(主要介绍一下豆瓣电影数据可视化项目的基本情况)

本项目主要实现将豆瓣电影网中的每年的电影数据—名称、评分、国家、年份等信息爬取到然后以柱状图、饼状图、折线图的形式展现出来。

# 需求分析

(主要说明一下要进行数据可视化提前需要准备的数据和文件资料等)

使用NodeJS编写爬虫来爬取豆瓣电影网的电影数据。需要一个MongoDB的数据库来存储爬取到的电影数据。

# 技术选型

(主要说明自己所负责的当前页面开发说使用的技术，以及每种技术的作用是干什么)

我主要负责爬取2011-2012年的所有电影数据，主要用nodeJS设置爬虫然后将数据传入到MongoDB的数据库中保存，然后将数据以JSON的文件形式导出。

# 爬虫开发和优化

(爬虫开发的步骤以及核心技术点)

首先建立数据库连接设置；再获取到豆瓣电影所有电影的URL；再通过遍历获取每一组url数据，根据每一组的url发送请求获取到HTML页面；再解析获取到的HTML并获取所需要的数据；判断数据是否重复；

# 数据清洗和存储

(对爬到的数据进行清洗和二次处理，处理完成保存到mongodb)

mongoose.connect('mongodb://127.0.0.1:27017/school', {

useNewUrlParser: true

})

let moviedata = mongoose.model('Movie', Movie);

async function GetData(url) {

return new Promise((reslove, reject) => {

let index = 0;

let i = 1;

let timer = setInterval(() => {

if (url[index] == undefined) {

clearInterval(timer);

reslove(undefined);

} else {

superagent.get(url[index++]).end((err, data) => {

if (err) {

console.log('页面' + (i++)+ '不存在');

} else {

let $ = cheerio.load(data.text);

let Yeardata = $('#content h1 span').last().text().split('');

let Year = Yeardata[1] + Yeardata[2] + Yeardata[3] + Yeardata[4];

//匹配国家de正则表达式

let reg = /制片国家\/地区:\s([A-z\u4e00-\u9fa5]+(\s\/\s[A-z\u4e00-\u9fa5]+)\*)/m;

let temp = {

name: $('#content h1 span').first().text(),

rate: $('.ll.rating\_num').text(),

coutury: reg.exec($('#info').text())[1],

year: Year,

comments: $('.rating\_people span').text()

}

//判断是否为空，为空调出

let boolea = true;

for (let attr in temp) {

if (temp[attr] == ' ') {

boolea = false;

}

}

if (boolea == false) {

console.log("第" + (i++)+ '条数据存在空值，不存入数据库');

} else {

//查询重复

moviedata.find(temp, (err, ret) => {

if (err) {

console.log('查询' + (i++) + '数据失败,故不存入数据库');

}

//如果不重复才存入

if (ret == '') {

let item = new moviedata(temp);

//存入数据

item.save((err, ret) => {

if (err) {

console.log('第' + (i++) + '数据不存在,但调用save存入失败');

} else {

console.log('成功存入第' + (i++) + '条数据');

}

})

} else {

console.log('第' + (i++) + '数据已存在');

}

})

}

}

})

}

}, 1000)

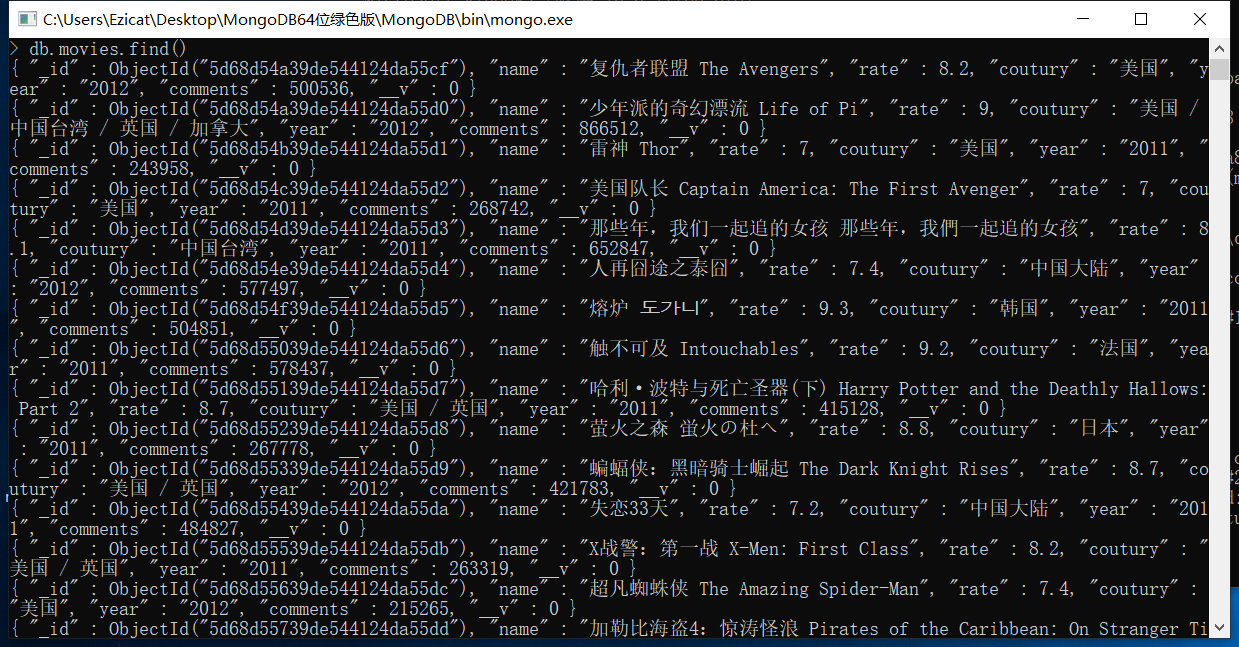
})

}

let movies = db.collection("movies")

# 数据查询和可视化展示

(根据不同条件在mongodb数据库查询对应的数据集，然后使用echarts进行可视化展示，生成<产品原型>部分的可视化图表)



# 开发环境

(主要说明在进行开发的时候的系统环境和开发IDE等，可以介绍一下选择这个IDE的原因)

VSCode，方便简洁，容易使用。

# 产品原型

(主要说明自己所负责的的模块最终的呈现效果)

