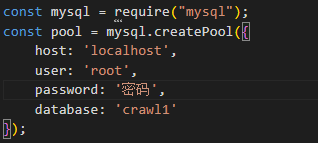
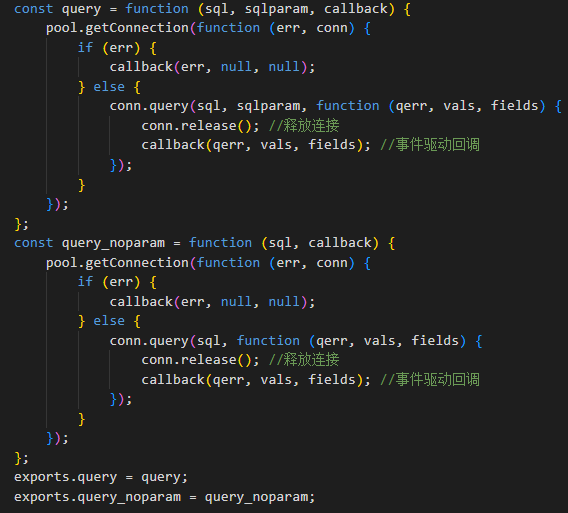
**简要说明一下两大板块（网页+爬虫）的代码实现过程**

**初始化数据库**

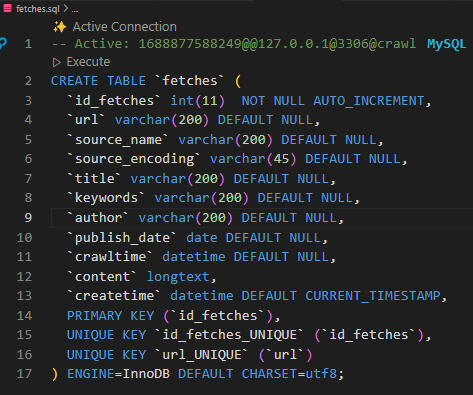
通过node.js连接mysql数据库：



封装的访问模块：



创建fetches表：



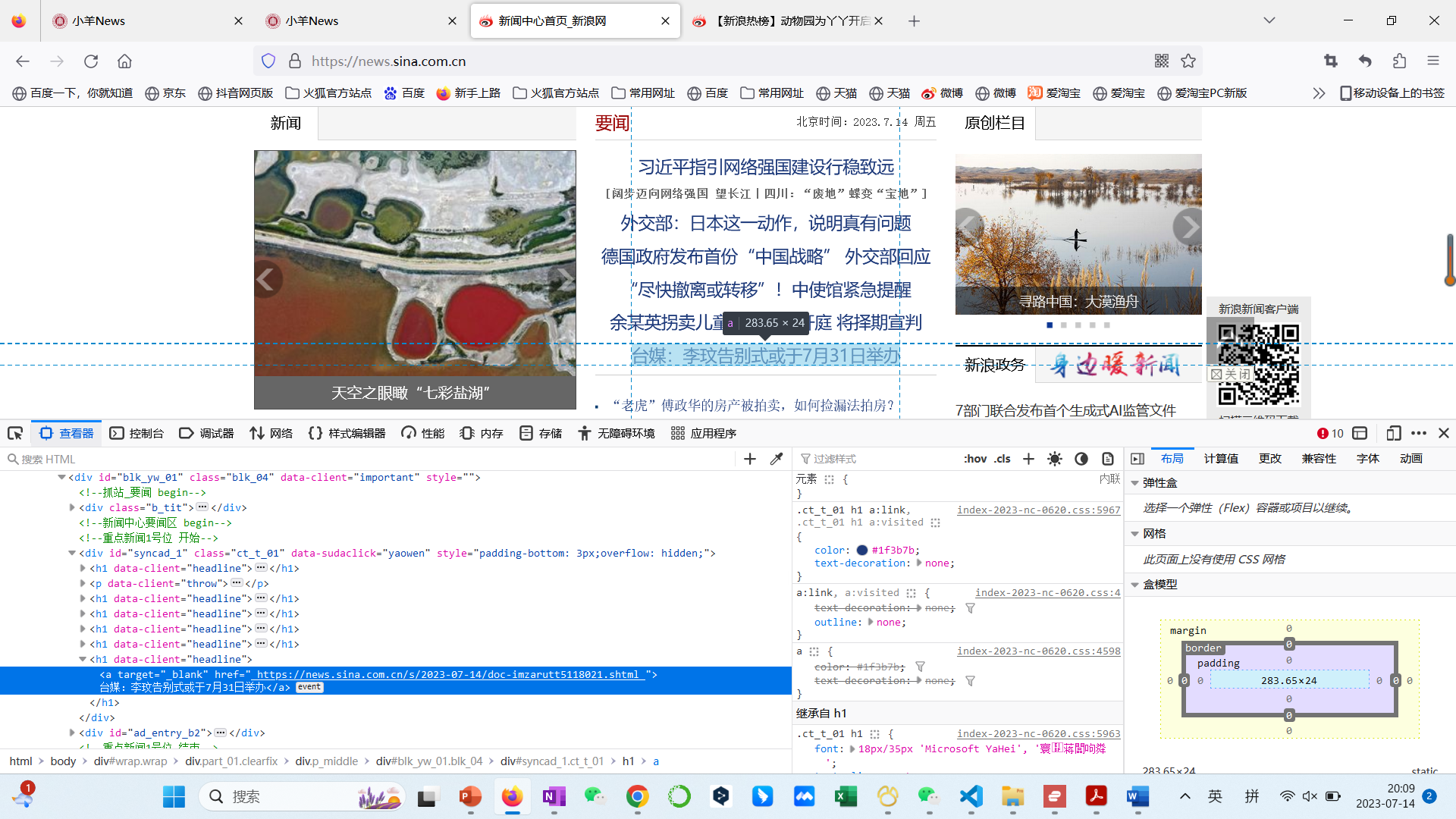
**爬虫部分**

爬虫板块我爬取了五个网站的新闻信息，分别是：

1. 新浪新闻 http://news.sina.com.cn/
2. 上海交通大学新闻学术网 https://news.sjtu.edu.cn/
3. 网易娱乐 https://ent.163.com/
4. 网易新闻 https://news.163.com/
5. 中国青年网 http://www.youth.cn/

下面以**新浪新闻**为例进行介绍，不逐一介绍全部的五个网站

⾸先分析将要爬取的新闻⽹⻚。在⽹站上右击“检查”和“查看页面源代码”可以查看要爬取的⽹⻚的源码，利⽤代码栏左上⾓的“查看器”按钮，在⽹⻚上选择想要审视的要素，就可以看到该要素对应的html源代码。利⽤这个⽅式，我们可以确定要爬取的内容的格式。



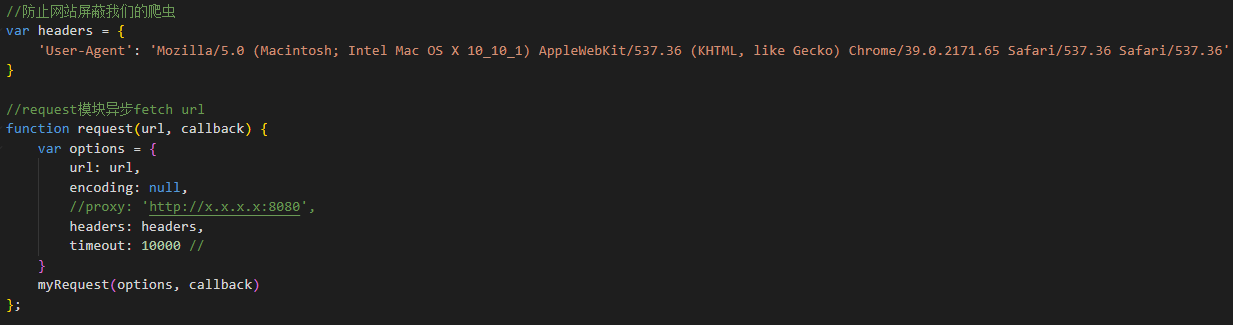


新浪新闻的主⻚是我们本次爬取的种⼦⻚⾯，我们将解析出这个⻚⾯上的新闻⻚⾯超链接并对这些新闻⻚⾯进⾏爬取，根据需要爬取的数据修改format内容，以及用正则表达式匹配新闻url。

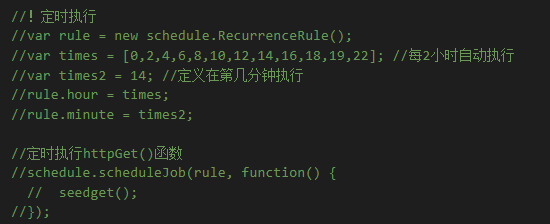
**使用的工具包：**

其中，request库是⼀个简单的客户端，⽤于发送http请求。cheerio库是为服务器端实现的⼀个快速敏捷的Jquery核⼼库，在这个项⽬⾥⽤来解析html。iconv-lite库可以将编码格式由GBK转为UTF-8，转换后的html就可以⽤cheerio解析。fs库是file system library的意思，是⽤来读取本地⽂件的。date-utils库⽤于转换⽇期格式。node-schedule库⽤来完成定时器功能。

**构建访问模块**



**定时执行**



**爬取种子页面**

在爬取种子页面时，本质上就是爬取新闻网页链接地址，供我们之后爬取新闻内容使用。网页链接一般在标签内。

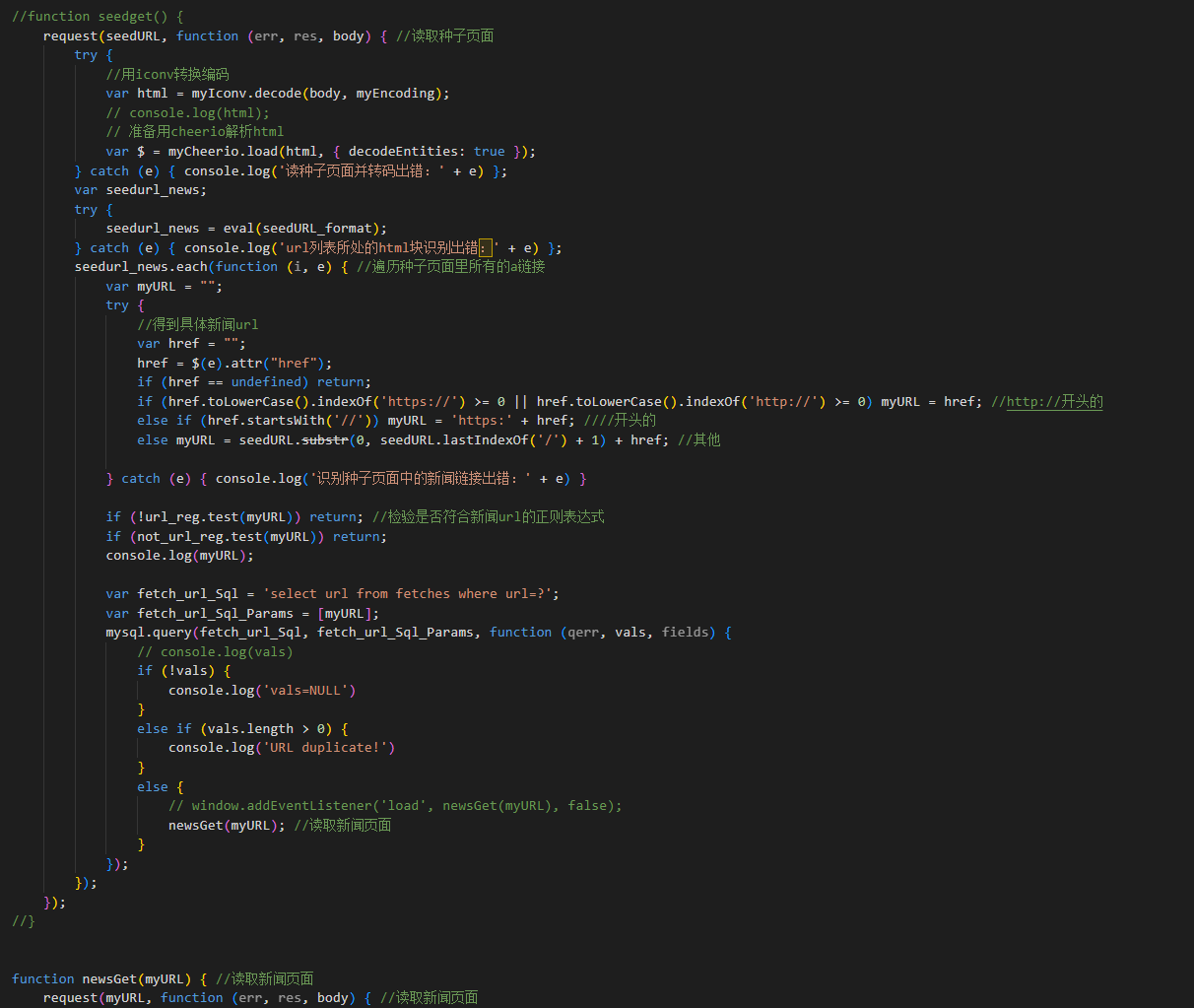
**重要的几点如下：**

1、使用\*\*seedURL\_format = "$('a')"\*\*选择所有的链接地址

2、检验新闻链接是否符合新闻URL的正则表达式，即判断改链接是否是一个有效的新闻地址(有可能出现404页面)。例如一个标准的新闻网址<https://news.sina.com.cn/gov/xlxw/2022-07-08/doc-imizirav2474084.shtml>

3、值得注意的是，链接有些是http，有些是https，需要分类处理；

4、对每个有效的链接发起request；



**爬取内容页面**

在内容页面，我们需要爬取的就是新闻具体内容，内容包括 title、author、keywords等。因为不同的内容位于不同的标签内，并且有不同的name，在之后我们要根据我们所需要爬取的内容，使用选择器选择我们要爬取的元素。爬取新浪新闻网需要用到的匹配format，通过选择器，我们可以选择我们想要获取的元素内容。

常用的选择器如下：

·元素选择器 $('title')

·类选择器 $('.class')

·id选择器 $('#author’)

·属性选择器 $('meta[name="author"]')



**newsGet函数**

该函数的作用为读取某一个有效URL的新闻内容，也是代码的核心部分。主要作用如下：

·用iconv转换编码

·用cheerio解析html\_news

·判断是否是有效页面

·通过选择器，读取元素内容

·匹配日期格式

·判断改页面是否已经被爬取过（查询数据库，判断数据库中是否已经存在）；

·写入数据库



**学习心得：**

在爬取不同的新闻页面时，代码的框架能够直接复用，但我们需要根据不同页面的实际情况，分析种子页面和新闻页面，使用选择器选择我们想要读取的html标签，从而获取内容。

**网页部分**

**网站框架**

·网站后端使用node.js express框架

·网站前端引入了少量bootstrap组件

**Express脚手架**

Express脚手架是一个规范的项目模版。通过使用脚手架，我们能够快速并且规范地使用express框架服务。

Express脚手架目录如下：

·bin/：www文件所在文件目录，我们通过运行www文件启动Express；

·public/：通常用来存放浏览器拉取的公共资源，如图片等；

·routes/：index.js所在文件目录，定义Express的响应路由；

·views：存放Express动态渲染的 HTML 页面，但在此文件后缀为.ejs，但作用与HTML 相同；

**Bootstrap**

Bootstrap 是一个用于快速开发 Web 应用程序和网站的前端框架，Bootstrap 包含了许多可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等，能够帮助我们快速搭建Web前端网页。

**Express网站后端**

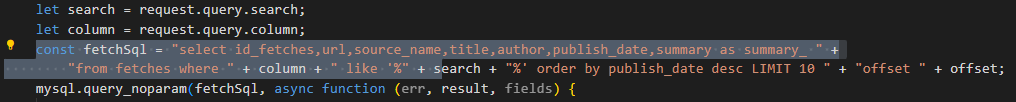
**数据库查询**

·选择查找的 column；

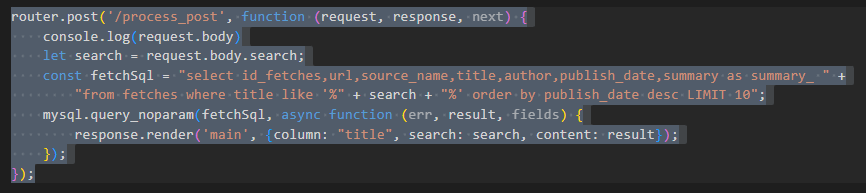
·选择要匹配like的关键词；

·选择升序和降序；

·选择LIMIT offset查找范围；



**请求路由**

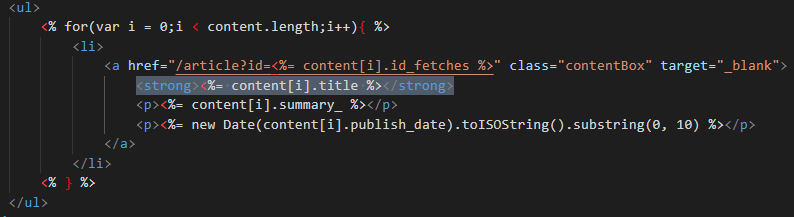


**网页渲染**

·通过 response.render 可以渲染HTML页面；

·控制语句<% ？？？ %>；

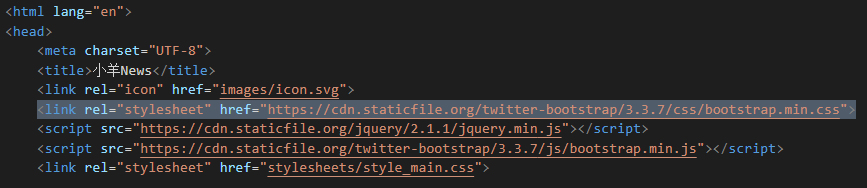
·渲染语句<%= ？？？ %>；



**网页前端**

**Boostrap组件**

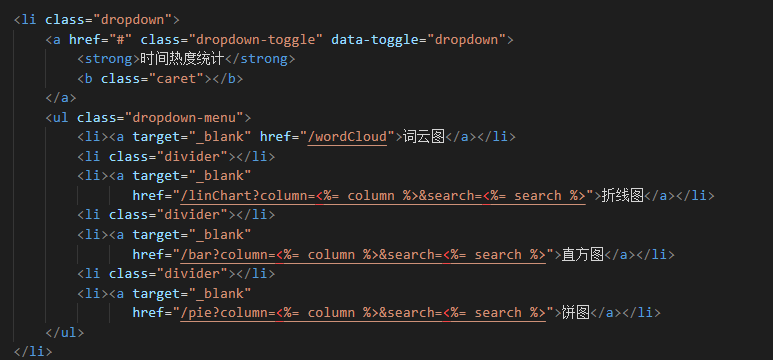
**组件引入**



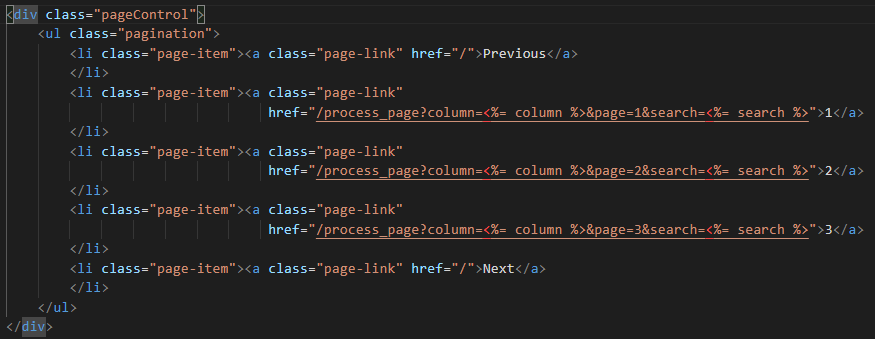
**导航栏**



**下拉菜单**

**分页管理**



**课上讲过的Echarts**

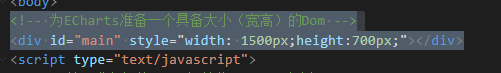
**使用Echarts绘制图表（四种图的代码区别不大）；**

·词云图（绘制词云图前，需要使用 node segment进行中文分词，并统计词频；）

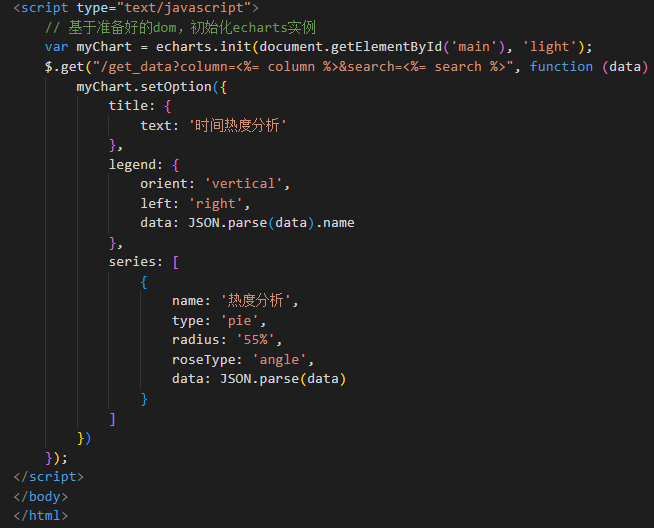
·折线图

·直方图

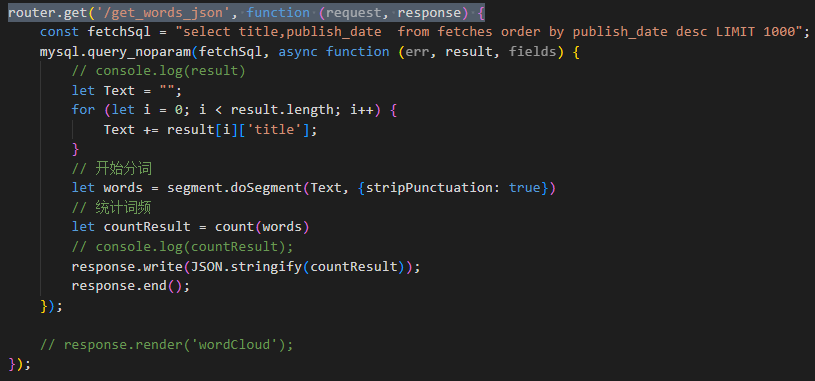
·饼图



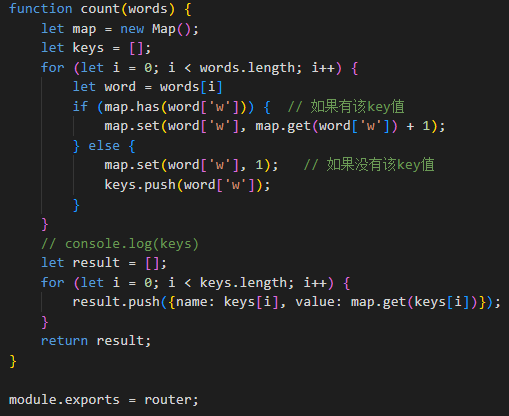
**异步加载**



**中文分词**



**统计词频**



**学习心得：**

在本次Project中，主要使用node.js和html进行实现。node.js中的express是一个使用非常方便的后端框架，并且express提供的脚手架已经能够被直接访问了，我们只需要添加新的路由信息和相应的处理函数即可快速搭建一个简单的Web网站，本次的Web网站前端由六个主要的html组成。