1. 学子问答--首页实践
2. 数据表结构

数据库名称：xzqa

xzqa\_category,存储文章的类型

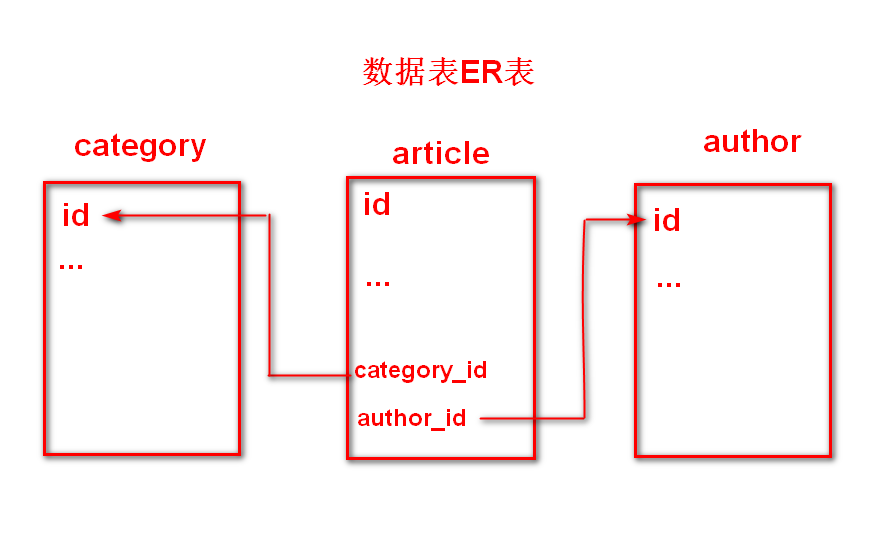
|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 描述 |
| id | SMALLINT | 文章类型的ID,主键且自增 |
| category\_name | VARCHAR | 文章类型的名称，唯一约束 |

xzqa\_author,存储文章的作者

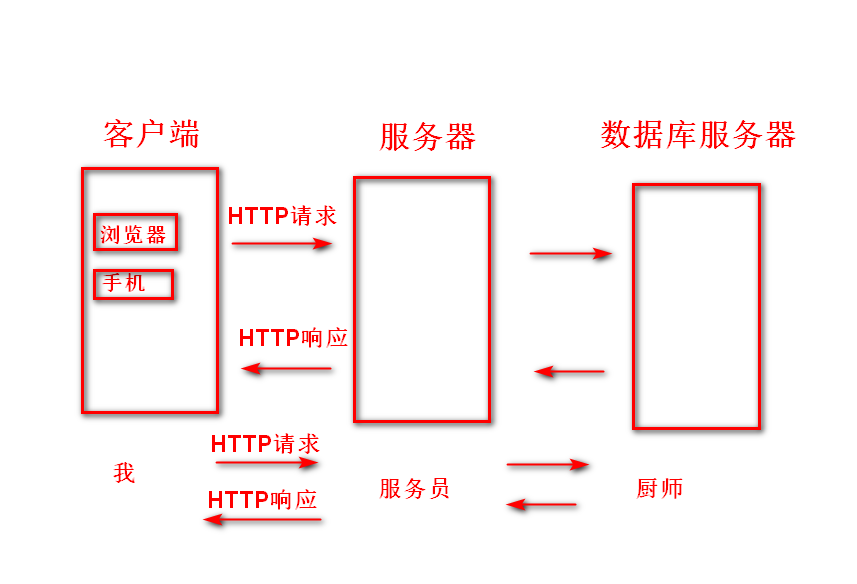
|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 描述 |
| id | MEDIUMINT | 作者ID,主键且自增 |
| username | VARCHAR | 用户名，唯一约束 |
| password | VARCHAR | 密码，MD5加密 |
| nickname | VARCHAR | 用户的昵称 |
| avatar | VARCHAR | 用户的头像，默认为值unnamed.jpg |
| article\_number | MEDIUMINT | 用户发表的文章数量，默认为值0 |

xzqa\_article,存储文章

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 描述 |
| id | INT | 文章ID，主键且自增 |
| subject | VARCHAR | 文章标题 |
| description | VARCHAR | 文章简介 |
| image | VARCHAR | 文章缩略图像,**可以为空** |
| content | MEDIUMTEXT | 文章正文(文章详细信息) |
| category\_id | SMALLINT | 文章分类ID,外键，参照文章分类表的ID |
| author\_id | INT | 作者ID,外键，参照作者表的ID |
| created\_at | INT | 文章的发表时间 |



1. WEB应用的工作流程



1. 创建WEB服务器
   1. 创建空白目录，并且在Windows命令行中切换到该目录
   2. 输入以下命令

|  |
| --- |
| npm install --save express mysql cors |

* 1. 在WEB服务器目录下创建app.js，并且输入以下代码：

|  |
| --- |
| //加载Express模块  const express = require('express');  //创建Express应用  const server = express();  server.listen(3000); |

* 1. 在VSCode编辑器的终端中输入

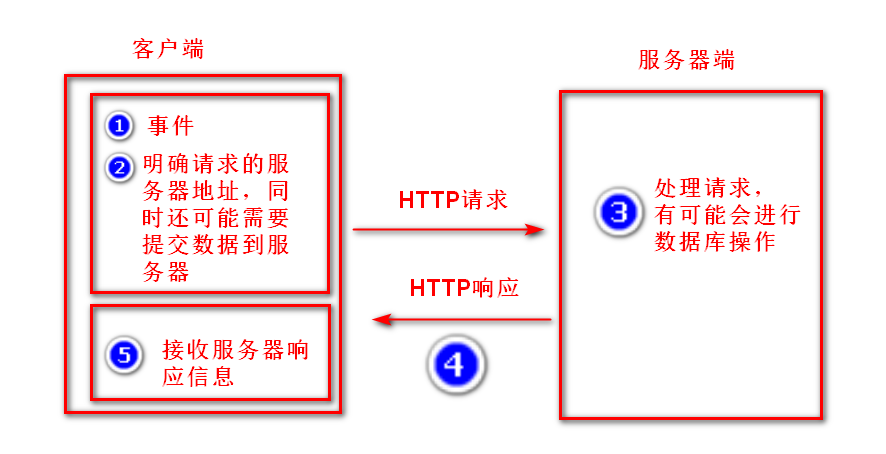
|  |
| --- |
| node app.js |

* 1. 启动浏览器，并且输入 http://127.0.0.1:3000

1. 客户端向服务器发送HTTP请求

安装Axios

|  |
| --- |
| npm install --save axios |



WEB服务器的修改流程：

第一步，ctrl+c，停止WEB服务器

第二步：根据需要修改服务器的相关配置

第三步：启动WEB服务器

1. 顶部选项卡及面板的实现

通过xzqa\_category(文章类型表)中的记录来决定顶部选项卡及面板的数量和内容，其涉及到的问题有：

* 什么事件下要发送AJAX请求到服务器，以获取相关的数据? -- mounted

Vue的生命周期及涉及的钩子函数，钩子函数有：

* beforeCreate
* created
* beforeMount
* mounted
* beforeUpdate
* updated
* beforeDestroy
* destroyed
* 发送请求的服务器地址是什么?

规定获取文章类型的服务器API规范如下：

路由地址为 /category，

请求方式为get

* WEB服务器如何获取到指定数据表中的数据?

现在需要通过MySQL模块的来完成MySQL数据的增删改查操作

//加载MySQL模块

const mysql = require('mysql');

//创建MySQL连接池

const pool = mysql.createPool({

     //MySQL数据库服务器的地址

     host:'127.0.0.1',

     //MySQL数据库服务器端口号

     port:3306,

     //用户名

     user:'root',

     //密码

     password:'',

     //数据库名称

     database:'xzqa'

});

而且可以通过MySQL连接池提高查询效率。所以现在需要在WEB上加载MySQL模块并且进行相关的配置。

同时还需要在WEB服务器上创建名称为"/category"的路由，并且获取xzqa\_category数据表的全部记录，同时将查询的结果返回到客户端

//获取文章分类信息的API

server.get('/category',(req,res)=>{

    //查询xzqa\_category(文章类型)表中所有的记录

    var sql = 'SELECT id,category\_name FROM xzqa\_category';

    // MySQL连接执行查询操作

    pool.query(sql,(err,results)=>{

        //如果发生异常,则直接抛出异常

        if(err) throw err;

        //将相关的查询信息返回到客户端

        res.send({message:'查询成功',code:200,category:results});

     });

});

* 当客户端接收到服务器返回的数据后，如何动态实现顶部选项卡及面板？

当客户端接收到服务器返回的结果后，必须存储到自身的变量 -- 此时在data中声明category变量，用于存储服务器返回的数据 ，才可以在页面组件中使用。

所以代码如下：

mounted(){

     this.axios.get('/category').then((res)=>{

        //将WEB服务器返回的文章分类信息存储到Vue的变量中

        this.category = res.data.category;

     });

   }

既然现在this.category已经是数组类型的数据了，那么就可以在<template>中通过循环结构来决定***顶部选项卡***的数量及内容了！-- 通过v-for指令实现，代码如下：

<mt-navbar v-model="active">

      <mt-tab-item

        v-for="(item,index) in category"

        :key="index"

        :id="item.id">

        {{item.category\_name}}

      </mt-tab-item>

    </mt-navbar>

此时通过category变量不仅要控制顶部选项卡的数量及内容，**还要控制面板的数量**,所以还需要对于<mt-tab-container-item>进行循环操作,代码如下：

<mt-tab-container v-model="active">

        <mt-tab-container-item

          v-for="(item,index) in category"

          :key="index"

          :id="item.id">

...

        </mt-tab-container-item>

      </mt-tab-container>

1. 文章列表的实现

涉及到两种情况:

第一种：**初始情况**下要获取默认**被选定选项卡**的文章信息

"初始情况"已经表明事件 -- 涉及Vue的生命周期及钩子函数 -- mounted

获取相关选项卡文章API的规定如下：

请求方式：get

路由地址：/getArticles

请求参数：cid,整数类型，代表文章分类ID

**强调URL地址栏传递的形式为：**

**URL地址?参数名称=值&参数名称=值**

此时客户端的代码如下：

  mounted(){

//获取默认选项卡包含的文章信息

    this.axios.get('/getArticles?cid=' + this.active)

  }

目前代表已经向WEB服务器发送请求了，所以WEB服务器需要存在对应的路由信息，代码如下：

//获取指定分类文章信息的API

server.get('/getArticles',(req,res)=>{

    //获取地址栏请求参数 -- cid(文章分类ID)

    var cid = req.query.cid;

    //以获取到的cid参数为条件查询该分类下的文章信息

    var sql = 'SELECT id,subject,description,image FROM xzqa\_article WHERE category\_id=?';

    //执行SQL查询

    pool.query(sql,[cid],(err,results)=>{

        if(err) throw err;

        res.send({message:'查询成功',code:200,articles:results});

    });

});

当成功运行后，服务器已经可以将查询结果返回给客户端，所以客户端需要接收数据，并且显示在页面组件中。

声明变量 -- articles用于存储WEB服务器返回的信息，代码如下：

  this.axios.get('/getArticles?cid=' + this.active).then((res)=>{

      //将WEB服务器返回的文章信息存储到articles数组中

      this.articles = res.data.articles;

  })

在页面组件中循环 articles 变量，以输出该分类下的所有文章信息，代码如下：

<div class="articleItem"

     v-for="(article,index) in articles"

     :key="index">

...

</div>

第二种：在切换顶部选项卡时，要获取当前选项卡的文章信息

事件：就是在切换顶部选项卡时进行请求的发送，而可以通过watch监听到active变量的变化即可，代码如下：

  active(){

        this.axios.get('/getArticles?cid=' + this.active).then((res)=>{

          this.articles = res.data.articles;

        })

    }

作业：

1.实现切换顶部选项卡时，也能读取出图像

2.当第一步实现完成后，可以会出现文章累加，怎么解决？

3.如何不显示没有图片的那个DIV

