

# Mint UI Unit03

## 1.CSS组件

### • TabContainer 组件

TabContainer 组件用于实现面板，其语法结构是：

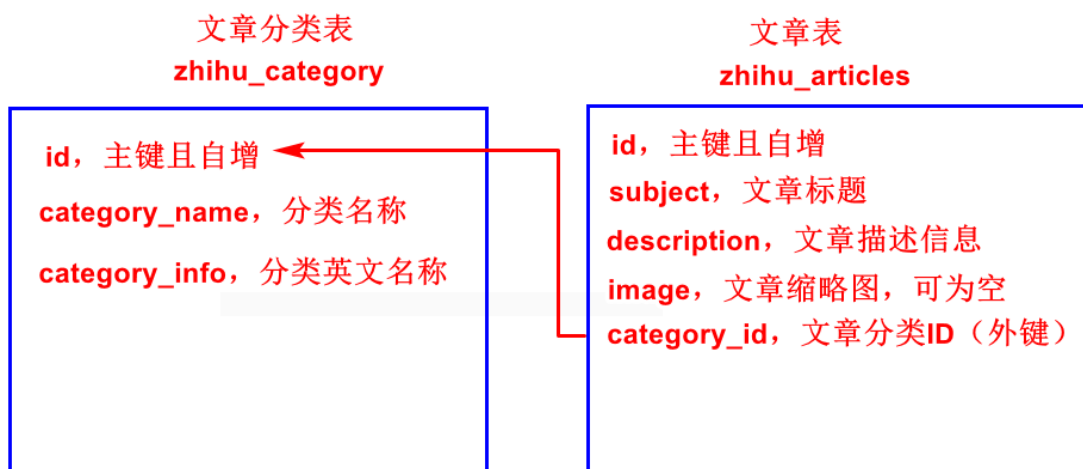
```
<mt-tab-container v-model="变量名称" swipeable>
  <mt-tab-container-item id="当前面板的ID">
    ...
  </mt-tab-container-item>
  <mt-tab-container-item id="当前面板的ID">
    ...
  </mt-tab-container-item>
  ...
</mt-tab-container>
```

v-model 变量的数据类型为字符类型，实质上为 mt-tab-container-item 的 ID

swipeable 属性控制面板在切换时是否显示滑动效果，布尔类型

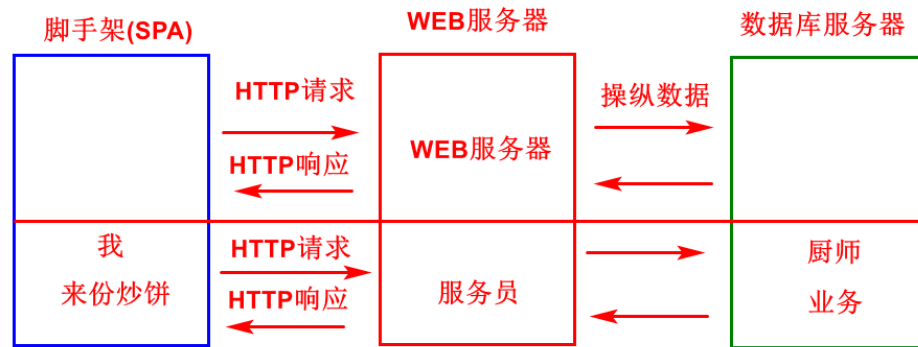
## 2.知乎首页的实现

### 2.1 数据表结构



### 2.2 获取首页数据(版本1)

1. 现在要获取首页的数据，而数据是存在于数据库服务器中的，所以现在需要连接WEB服务器，并且通过WEB服务器去获取数据库服务器的数据。流程如下：



2. 问题1--请问在什么情况下要获取服务器的数据？

在页面加载成功后就可获取服务器的数据！！引发问题--vue 的生命周期及钩子函数

- o beforeCreate
- o created
- o beforeMount
- o mounted
- o beforeUpdate
- o updated
- o beforeDestroy
- o destroyed

在本案例中采用 mounted 钩子函数进行获取数据的操作

既然要发送 AJAX 请求，所以必须使用 axios 完成，故修改 main.js 中关于 axios 的相关配置：

- o 安装 axios

```
npm install --save axios
```

- o 在 main.js 中对 axios 进行相关的配置

```
import axios from 'axios'

axios.defaults.baseURL = 'http://127.0.0.1:4000'

Vue.prototype.axios = axios;
```

baseURL 地址中的 4000 端口是后续WEB服务器的端口号（暂时没有）

现在已经完成 axios 的安装与配置，现在就可以在 mounted 钩子函数中发送 AJAX 请求了，代码截图如下：

```
JS index.js | Home.vue | JS main.js
src > views > Home.vue > {} "Home.vue" > script > mounted

41   margin-top:40px;
42 }
43 </style>
44 <script>
45 export default {
46   data(){
47     return {
48       selected:'recommend'
49     }
50   },
51   mounted(){
52     //向WEB服务器发送AJAX请求，以获取数据
53     this.axios.get('/articles');
54   }
55 }
56 </script>
```

但此时的运行结果截图如下：

```
[HMR] Waiting for update signal from WDS... log.js?1afd:24
Download the Vue Devtools vue.runtime.esm.js?2b0e:8429
extension for a better development experience:
https://github.com/vuejs/vue-devtools

✖ ▶ GET http://127.0.0.1:4000/articles xhr.js?b50d:178
net::ERR_CONNECTION_REFUSED

✖ ▶ Uncaught (in promise) Error: Network createError.js?2d83:16
Error
    at createError (createError.js?2d83:16)
    at XMLHttpRequest.handleError (xhr.js?b50d:83)

✖ ▶ GET http://127.0.0.1:4000/articles xhr.js?b50d:178
net::ERR_CONNECTION_REFUSED

✖ ▶ Uncaught (in promise) Error: Network createError.js?2d83:16
Error
    at createError (createError.js?2d83:16)
    at XMLHttpRequest.handleError (xhr.js?b50d:83)

>
```

此时出现如上图所示的错误，其根本原因是 -- 目前没有WEB服务器!!!

### 3. 问题2 -- 创建WEB服务器

安装 Node.js 的 Express 服务器、MySQL 中间件等

```
npm install --save express mysql
```

创建 app.js，代码如下：

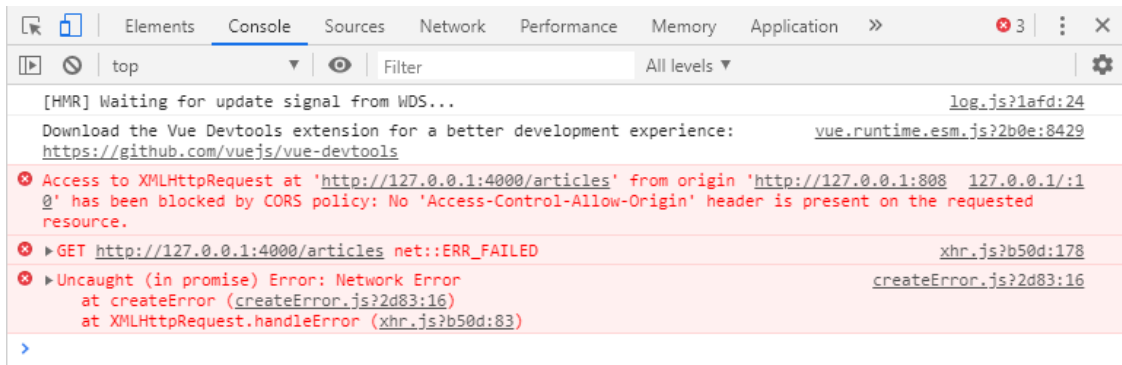
```
//加载Express模块
const express = require('express');

//创建Express应用
const server = express();

//监听端口
server.listen(4000);
```

在vscode中, 右击 app.js 中选择在"终端打开", 然后输入 `node app.js`

此时再查看脚本架的运行结果截图如下:



该错误是由于跨域操作引起的, 分析:

脚手架的地址是: <http://127.0.0.1:8080> 或 <http://localhost:8080>

WEB服务器的地址是: <http://127.0.0.1:4000> 或 <http://localhost:4000>

端口号不同, 所以属于跨域!!!

所以还必须在 Node.js 中安装 cors 模块, 以解决跨域!

在vscode中 ctrl+C, 停止WEB服务器

安装 cors 模块

```
npm install --save cors
```

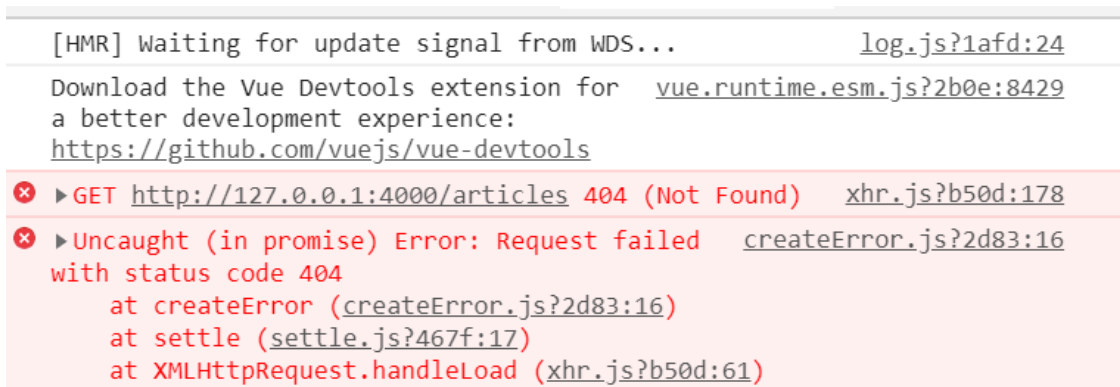
在 app.js 中对于 cors 进行相关的配置, 代码如下:

```
const cors = require('cors');

server.use(cors({
  origin: ['http://localhost:8080', 'http://127.0.0.1:8080']
}));
```

在vscode的终端中输入 `node app.js`

此时脚手架的运行结果截图如下:



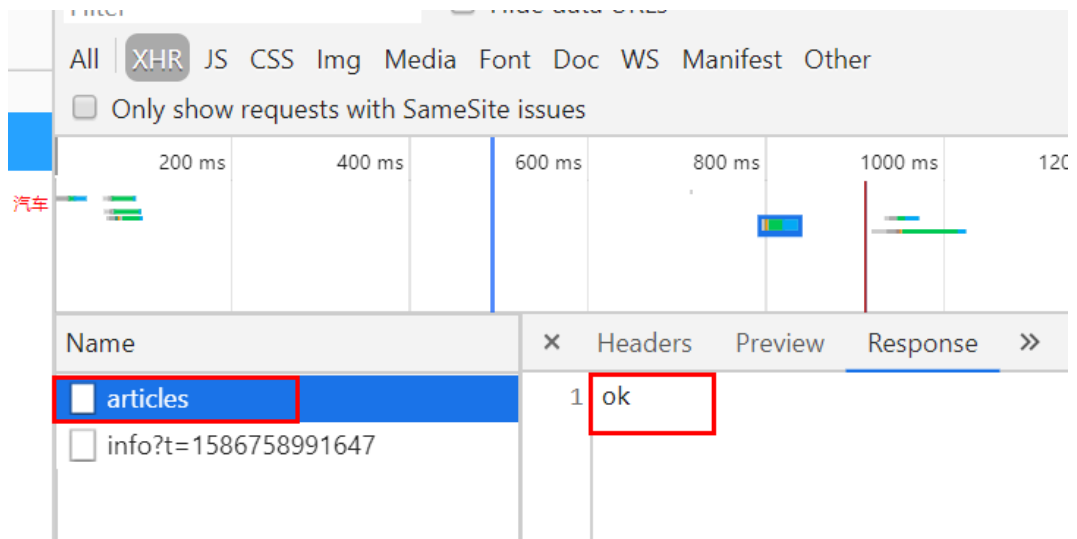
出现以上错误的原因是：在WEB服务器上不存在指定的路由

#### 4. 问题3 -- 在WEB服务器上创建指定的路由

- 第一步：在vsCode中`ctrl+c`，停止WEB服务器
- 第二步：配置路由，代码如下：

```
//articles的GET路由
server.get('/articles', (req, res) => {
  res.send('ok');
});
```

- 第三步：在vsCode的终端中输入`node app.js`，此时脚手架的运行结果截图如下：



#### 5. 问题4，在服务器的 /articles 请求中，要获取数据库服务器的数据，并且返回客户端，操作步骤如下：

- 第一步：在vsCode中`ctrl+c`，停止WEB服务器
- 第二步：通过MySQL连接池获取数据库的相关记录，代码如下：

```
//加载MySQL模块并且进行配置
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,
    //数据库用户的用户名
    user: 'root',
    //数据库用户的密码
    password: '',
    //数据库名称
    database: 'zhihu',
    //数据库编码方式
    charset: 'utf8',
    //最大连接数
    connectionLimit: 15
  });

```

在 `/articles` 路由中, 要通过 `MySQL` 连接池, 查找 `zhihu_articles` 数据表的"全部"记录, 代码如下:

```

server.get('/articles', (req, res) => {
  //查找 zhihu_articles(文章表)的"全部"记录
  var sql = 'SELECT * FROM zhihu_articles';

  //执行SQL查找操作
  pool.query(sql, (err, results) => {
    //如果发生错误,则直接抛出错误
    if (err) throw err;

    res.send({ message: '查询成功', code: 1, results: results });
  });
});

```

- 第三步: 在 `vscode` 的终端中输入 `node app.js`

6. 问题5: 既然现在WEB服务器已经成功返回数据, 那么必须在原来的发送 `AJAX` 的请求中接收服务器返回的数据, 然后在页面中显示。代码如下:

```

//src/views/Home.vue
mounted(){
  //向WEB服务器发送AJAX请求, 以获取数据
  //此时代表向WEB服务器articles路由发送GET请求, 并且接收服务器返回的数据
  this.axios.get('/articles').then(res=>{
    console.log(res.data.results);
  });
}

```

但是这些数据要显示的页面上, 所以必须通过一个属性来存储这些从WEB服务器返回的信息, 然后在页面中再通过 `v-for` 指令进行循环即可, 代码如下:

```

<script>
export default {
  data(){
    return {
      selected: 'recommend',
      //存储服务器返回的信息
      data:[],
    },
  },
  mounted(){
    //向WEB服务器发送AJAX请求，以获取数据
    //此时代表向WEB服务器articles路由发送GET请求,并且接收服务器返
    this.axios.get('/articles').then(res=>{
      //将服务器返回的信息存储到data属性中
      this.data = res.data.results;
      //console.log(this.data);
    });
  }
}

```

```

<mt-tab-container v-model="selected" swipeable>
  <mt-tab-container-item id="recommend">
    <p v-for="(item,index) in data" :key="index">{{item.subject}}</p>
  </mt-tab-container-item>
  <mt-tab-container-item id="life">
    <p>生活新闻</p>
  </mt-tab-container-item>
  <mt-tab-container-item id="entertainment">
    <p>娱乐新闻</p>
  </mt-tab-container-item>
  <mt-tab-container-item id="automobile">
    <p>汽车新闻</p>
  </mt-tab-container-item>

```

## 2.3 获取首页数据(版本2)

虽然，版本1中获取了数据，但是在进行导航切换时，并不能实现获取对应的数据。

也就意味着在获取数据时，需要明确告知服务器---所需要的数据的类型是什么！分成两种情况解决：

### 1. 初始情况下：

操作步骤如下：

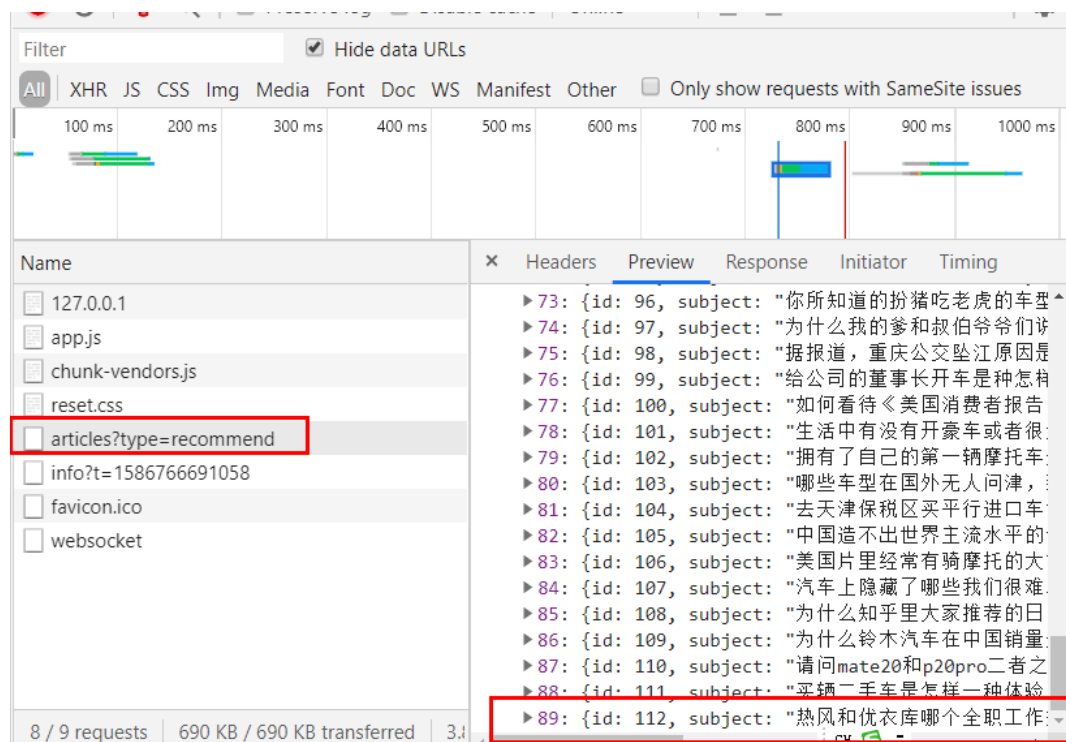
- 修改原 `src/views/Home.vue` 中的 `mounted` 钩子函数，其必须传递给WEB服务器对应的文章分类，代码截图如下：

```
mounted(){
  //向WEB服务器发送AJAX请求, 以获取数据
  //此时代表向WEB服务器articles路由发送GET请求,并且接收服务器返回的数据
  this.axios.get('/articles?type=' + this.selected).then(res=>{
    //将服务器返回的信息存储到data属性中
    this.data = res.data.results;
    //console.log(this.data);
  });
}
```

友情提示: URL地址栏传递的形态为:

URL?参数名称=值&参数名称=值&参数名称=值

此时的运行结果如下图所示:



虽然请求参数已经成功发送,但是服务器返回的数据根本没有任何变化--原因是服务器还没有接收处并使用参数,所以:

第一步: 在vsCode中 `ctrl+C`, 停止WEB服务器

第二步: 代码如下:

```
//articles的GET路由
server.get('/articles', (req, res) => {

  //获取URL地址中名称为type的参数
  var type = req.query.type;
  console.log(type);
  //查找 zhihu_articles(文章表)的"全部"记录
  var sql = 'SELECT * FROM zhihu_articles';

  //执行SQL查找操作
  pool.query(sql, (err, results) => {
```



```

//如果发生错误,则直接抛出错误
if (err) throw err;

res.send({ message: '查询成功', code: 1, results: results });

});
});

```

此时，可以在服务器的终端中获取到当前分类名称，截图如下：



但问题是，在 `zhihu_articles` 这张数据表中只存储到分类的ID，并非分类的名称！！

**所以需要多表连接操作**

此时的 SQL 需要修改为：

```

var sql = 'SELECT zhihu_articles.id,subject,description,image FROM
zhihu_articles,zhihu_category WHERE category_id = zhihu_category.id AND
category_info=?';

```

既然 SQL 中存在占位符，那么在执行时，必须为占位符赋予有效值，所以 `query` 方法修改为：

```

pool.query(sql, [type],(err, results) => {
//如果发生错误,则直接抛出错误
if (err) throw err;
res.send({ message: '查询成功', code: 1, results: results });

});

```

第三步：在 `vsCode` 的终端中输入 `node app.js`

2. 单击顶部导航的情况

在单击顶部导航时，同样要获取该类型下的文章，而切换顶部导航时可以引发 `selected` 属性的变化，所以只需要监听该变量的值，然后再向服务器发送请求即可。

代码如下：

```
watch:{
  selected(){
    //发送AJAX请求
    this.axios.get('/articles?type=' + this.selected).then(res=>{
      console.log(res.data.results);
    });
  }
}
```

此时在切换顶部选项卡时，确实浏览器的控制台输出相应分类下的文章信息了，但是其并没有显示在对应的面板中。

按照常理说，应该再声明三个类似于 `data` 的变量，用于分别存储不同分类下的文章信息，如果分类越多，变量的命名越复杂，最好的解决方案是：用 `data` 存储所有不同分类下的数据，结果如下：

```
selected: 'recommend',
data: {
  //存储推荐新闻
  recommend: [],
  //存储生活新闻
  life: [],
  //存储娱乐新闻
  entertainment: [],
  //存储汽车新闻
  automobile: []
}
```

//之所有将`data`对象的成员的命名与选项卡名称、面板名称相同的原因是：我们想通过一个变量来控制这三个不同的信息，因为只要`selected`属性发生变化，也就意味着顶部导航发生变化，面板发生变化，数据的循环发生变化！！