

主页功能开发

Flex布局

采用Flex布局的元素，称为Flex 容器，简称容器。它的所有子元素自动成为容器成员，称为Flex项（Flex item）。

Flex布局有2个轴，主轴和侧轴（cross，交叉轴）。

默认X轴为主轴，从左向右；侧轴Y轴，垂直向下。默认线性沿着主轴排列，不换行。

- justify-content 设置主轴上子元素排列方式
 - flex-start 默认值，沿着主轴排列。如果主轴是row，子元素从左至右排列
 - flex-end 从右开始对齐
 - center 子元素居中对齐
 - space-around 平分剩余空间
 - **space-between** 先两边贴边，再平分剩余空间（重要）
- align-items 设置侧轴上的子元素的排列方式（单行）
 - 控制一行元素侧轴上的排列效果
 - flex-start 从头开始
 - flex-end 从尾开始
 - **center** 居中
 - stretch 沿着侧轴拉伸，子元素不要给高度

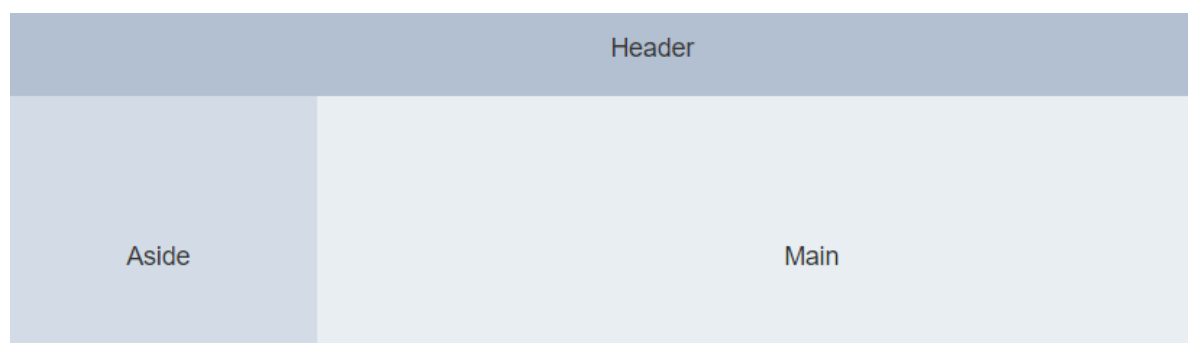
div a中增加2个div b和c，默认div是block，所以这b、c是各占一行。a采用flex后，b、c称为Flex item，并挤在一起了。默认所有子元素按照行从左开始排列，默认不折行。

主页布局

背景颜色可以参考 <https://preview.pro.ant.design/dashboard/analysis>

参考 <https://element.eleme.cn/#/zh-CN/component/container>，选择一个合适的布局

```
1 <el-container>
2   <el-header>Header</el-header>
3   <el-container>
4     <el-aside width="200px">Aside</el-aside>
5     <el-main>Main</el-main>
6   </el-container>
7 </el-container>
```



主页组件

src/plugins/element.js

```
1 import Vue from 'vue'
2 import { Form, FormItem, Input, Button, Message, Container, Header, Aside,
  Main } from 'element-ui'
3 import 'element-ui/lib/theme-chalk/index.css' // UI组件样式
4
5 Vue.use(Form)
6 Vue.use(FormItem)
7 Vue.use(Input)
8 Vue.use(Button)
9 Vue.use(Container)
10 Vue.use(Header)
11 Vue.use(Aside)
12 Vue.use(Main)
13
14 // 全局导入
15 Vue.prototype.$message = Message
```

src/compents/Home.vue

在Header中增加2个div，一个放logo，一个放退出按钮

```
1 <template>
2   <el-container>
3     <el-header>
4       <div class="logo">
5         
6         <div class="title">马哥教育猛犸运维系统管理平台</div>
7       </div>
8       <div class="info">
9         <el-button type="info" @click="logout">退出</el-button>
10      </div>
11    </el-header>
12    <el-container>
13      <el-aside width="200px">Aside</el-aside>
14      <el-main>Main</el-main>
15    </el-container>
16  </el-container>
17 </template>
18
19 <script>
20 export default {
21   methods: {
22     logout() {
23       window.localStorage.removeItem('token')
24       this.$router.push('/login')
25     }
26   }
27 }
28 </script>
29
30 <style lang="less" scoped>
```

```

31 .el-container {
32   height: 100%;
33 }
34
35 .el-header {
36   display: flex;
37   justify-content: space-between; // 主轴两端对齐
38   align-items: center; // 侧轴居中
39   padding-left: 5px;
40   .logo {
41     display: flex;
42     img {
43       width: 30px;
44     }
45     .title {
46       font-size: 24px;
47       margin-left: 5px;
48     }
49   }
50 }
51
52 .el-aside {
53   background-color: #112233;
54 }
55
56 .el-main {
57   background-color: #f0f2f5;
58 }
59 </style>

```

侧边栏菜单布局

src/plugins/element.js

```

1  import Vue from 'vue'
2  import { Form, FormItem, Input, Button, Message, Container, Header, Aside,
3  Main, Menu, MenuItem, Submenu } from 'element-ui'
4
5  import 'element-ui/lib/theme-chalk/index.css' // UI组件样式
6
7  Vue.use(Form)
8  Vue.use(FormItem)
9  Vue.use(Input)
10 Vue.use(Button)
11 Vue.use(Container)
12 Vue.use(Header)
13 Vue.use(Aside)
14 Vue.use(Main)
15 Vue.use(Menu)
16 Vue.use(MenuItem)
17 Vue.use(Submenu)
18
19 // 全局导入
20 Vue.prototype.$message = Message

```

侧栏导航菜单 <https://element.eleme.cn/#/zh-CN/component/menu#ce-lan>

基本嵌套结构

```
1 <el-menu default-active="2-2">
2   <el-submenu index="1">
3     <template slot="title"><i class="el-icon-menu"></i>导航一</template>
4     <el-menu-item index="1-1">修改密码</el-menu-item>
5     <el-menu-item index="1-2">用户列表</el-menu-item>
6   </el-submenu>
7   <el-submenu index="2">
8     <template slot="title"><i class="el-icon-menu"></i>导航二</template>
9     <el-menu-item index="2-1">修改密码</el-menu-item>
10    <el-menu-item index="2-2">用户列表</el-menu-item>
11  </el-submenu>
12 </el-menu>
```

src/components/Home.vue

```
1 <template>
2   <el-container>
3     <el-header>
4       <div class="logo">
5         
6         <div class="title">马哥教育猛犸运维系统管理平台</div>
7       </div>
8       <div class="info">
9         <el-button type="info" @click="logout">退出</el-button>
10      </div>
11    </el-header>
12    <el-container>
13      <el-aside width="320px">
14        <el-menu default-active="2-2" background-color="#123" text-
15        color="#fff" active-text-color="#ffd04b">
16          <el-submenu index="1">
17            <template slot="title"><i class="el-icon-menu"></i>用户管理
18            </template>
19            <el-menu-item index="1-1">用户列表</el-menu-item>
20            <el-menu-item index="1-2">用户列表</el-menu-item>
21          </el-submenu>
22          <el-submenu index="2">
23            <template slot="title"><i class="el-icon-menu"></i>导航二
24            </template>
25            <el-menu-item index="2-1">用户列表</el-menu-item>
26            <el-menu-item index="2-2">用户列表</el-menu-item>
27          </el-submenu>
28        </el-menu>
29      </el-aside>
30      <el-main>Main</el-main>
31    </el-container>
32  </el-container>
33 </template>
34
35 <script>
36 export default {
```

```

34     methods: {
35       logout() {
36         window.localStorage.removeItem('token')
37         this.$router.push('/login')
38       }
39     }
40   }
41 </script>
42
43 <style lang="less" scoped>
44   .el-container {
45     height: 100%;
46   }
47
48   .el-header {
49     display: flex;
50     justify-content: space-between; // 主轴两端对齐
51     align-items: center; // 侧轴居中
52     padding-left: 5px;
53     .logo {
54       display: flex;
55       img {
56         width: 30px;
57       }
58       .title {
59         font-size: 24px;
60         margin-left: 5px;
61       }
62     }
63   }
64
65   .el-aside {
66     background-color: #123;
67   }
68
69   .el-main {
70     background-color: #f0f2f5;
71   }
72 </style>

```

左边栏菜单内容可以客户端写死，也可以从服务端动态获取。

从服务端获取，可以先认证用户后，根据用户的权限，返回菜单项。

axios拦截器

前端通过token判断是否登录，但是更多的数据是服务器端返回的，当浏览器端发请求到服务器端，服务器端必须能认证身份，那么浏览器端必须在请求的Header中增加Authorization字段。

策略：只要向服务器端发请求，想获取数据，就必须进行token认证。

src/main.js

```

1 import Vue from 'vue'
2 import App from './App.vue'
3 import router from './router'

```

```
4 import './plugins/element.js'
5 import './assets/css/main.css'
6 import axios from 'axios'
7
8 // axios全局设置
9 // 全局拦截器，只要发起异步请求都要带上这个token
10 axios.interceptors.request.use(config => {
11   config.headers.Authorization = 'Bearer ' +
12   window.localStorage.getItem('token')
13   return config
14 })
15 // baseURL指向后台服务
16 axios.defaults.baseURL = '/api/v1/' // 代理到'http://127.0.0.1:8000/'
17 // 为vue类增加全局属性$http，这样所有组件实例都可以使用该属性了
18 vue.prototype.$http = axios
19
20 vue.config.productionTip = false
21
22 new Vue({
23   router,
24   render: h => h(App)
25 }).$mount('#app')
```

