

案例

需求

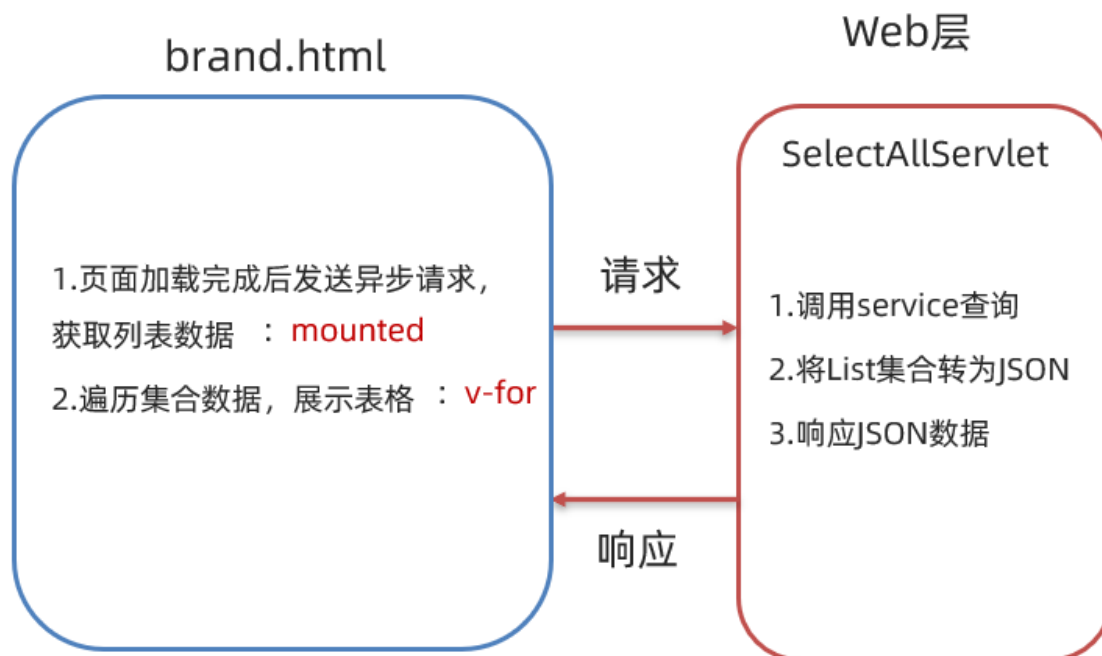
使用 Vue 简化我们在前一天ajax学完后做的品牌列表数据查询和添加功能

新增

序号	品牌名称	企业名称	排序	品牌介绍	状态	操作
1	三只松鼠	三只松鼠	100	三只松鼠, 好吃不上火	启用	修改 删除
2	优衣库	优衣库	10	优衣库, 服适人生	禁用	修改 删除
3	小米	小米科技有限公司	1000	为发烧而生	启用	修改 删除

此案例只是使用 Vue 对前端代码进行优化, 后端代码无需修改。

查询所有功能



1. 在 brand.html 页面引入 vue 的js文件

```
<script src="js/vue.js"></script>
```

2. 创建 Vue 对象

- 在 Vue 对象中定义模型数据
- 在钩子函数中发送异步请求, 并将响应的数据赋值给数据模型

```
new Vue({
  el: "#app",
  data(){
    return{
      brands: []
    }
  },
  mounted(){
    // 页面加载完成后, 发送异步请求, 查询数据
```

```

        var _this = this;
        axios({
            method: "get",
            url: "http://localhost:8080/brand-demo/selectAllServlet"
        }).then(function (resp) {
            _this.brands = resp.data;
        })
    }
})

```

3. 修改视图

- 定义 `<div id="app"></div>`，指定该 `div` 标签受 Vue 管理
- 将 `body` 标签中所有的内容拷贝作为上面 `div` 标签中
- 删除表格的多余数据行，只留下一个
- 在表格中的数据行上使用 `v-for` 指令遍历

```

<tr v-for="(brand,i) in brands" align="center">
    <td>{{i + 1}}</td>
    <td>{{brand.brandName}}</td>
    <td>{{brand.companyName}}</td>
    <td>{{brand.ordered}}</td>
    <td>{{brand.description}}</td>
    <td>{{brand.statusStr}}</td>
    <td><a href="#">修改</a> <a href="#">删除</a></td>
</tr>

```

整体页面代码如下：

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<div id="app">
    <a href="addBrand.html"><input type="button" value="新增"></a><br>
    <hr>
    <table id="brandTable" border="1" cellspacing="0" width="100%">
        <tr>
            <th>序号</th>
            <th>品牌名称</th>
            <th>企业名称</th>
            <th>排序</th>
            <th>品牌介绍</th>
            <th>状态</th>
            <th>操作</th>
        </tr>
        <!--
            使用v-for遍历tr
        -->
        <tr v-for="(brand,i) in brands" align="center">
            <td>{{i + 1}}</td>
            <td>{{brand.brandName}}</td>

```

```

        <td>{{brand.companyName}}</td>
        <td>{{brand.ordered}}</td>
        <td>{{brand.description}}</td>
        <td>{{brand.statusStr}}</td>
        <td><a href="#">修改</a> <a href="#">删除</a></td>
    </tr>
</table>
</div>
<script src="js/axios-0.18.0.js"></script>
<script src="js/vue.js"></script>

<script>
    new Vue({
        el: "#app",
        data(){
            return{
                brands: []
            }
        },
        mounted(){
            // 页面加载完成后，发送异步请求，查询数据
            var _this = this;
            axios({
                method: "get",
                url: "http://localhost:8080/brand-demo/selectAllServlet"
            }).then(function (resp) {
                _this.brands = resp.data;
            })
        }
    })
</script>
</body>
</html>

```

添加功能

页面操作效果如下：

添加品牌

品牌名称:

企业名称:

排序:

描述信息:

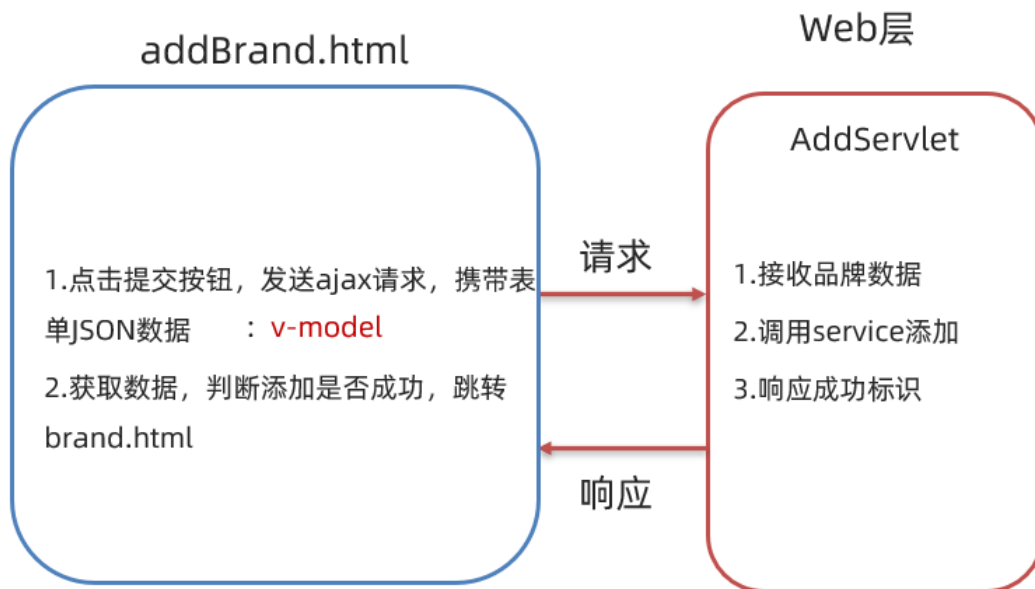
状态: ☐ 禁用 ☐ 启用

➔

新增

序号	品牌名称	企业名称	排序	品牌介绍	状态	操作
1	三只松鼠	三只松鼠	100	三只松鼠, 好吃不上火	启用	修改 删除
2	优衣库	优衣库	10	优衣库, 服适人生	禁用	修改 删除
3	小米	小米科技有限公司	1000	为发烧而生	启用	修改 删除

整体流程如下



==注意：前端代码的关键点在于使用 `v-model` 指令给标签项绑定模型数据，利用双向绑定特性，在发送异步请求时提交数据。==

1. 在 `addBrand.html` 页面引入 vue 的js文件

```
<script src="js/vue.js"></script>
```

2. 创建 Vue 对象

- 在 Vue 对象中定义模型数据 `brand`
- 定义一个 `submitForm()` 函数，用于给 提交 按钮提供绑定的函数
- 在 `submitForm()` 函数中发送 ajax 请求，并将模型数据 `brand` 作为参数进行传递

```
new Vue({
  el: "#app",
  data(){
    return {
      brand:{}
    }
  },
  methods:{
    submitForm(){
      // 发送ajax请求，添加
      var _this = this;
      axios({
        method:"post",
        url:"http://localhost:8080/brand-demo/addServlet",
        data:_this.brand
      }).then(function (resp) {
        // 判断响应数据是否为 success
        if(resp.data == "success"){
          location.href = "http://localhost:8080/brand-
demo/brand.html";
        }
      })
    }
  }
})
```

3. 修改视图

- 定义 `<div id="app"></div>`，指定该 `div` 标签受 Vue 管理
- 将 `body` 标签中所有的内容拷贝作为上面 `div` 标签中
- 给每一个表单项标签绑定模型数据。最后这些数据要被封装到 `brand` 对象中

```
<div id="app">
  <h3>添加品牌</h3>
  <form action="" method="post">
    品牌名称: <input id="brandName" v-model="brand.brandName"
name="brandName"><br>
    企业名称: <input id="companyName" v-model="brand.companyName"
name="companyName"><br>
    排序: <input id="ordered" v-model="brand.ordered" name="ordered">
<br>
    描述信息: <textarea rows="5" cols="20" id="description" v-
model="brand.description" name="description"></textarea><br>
    状态:
      <input type="radio" name="status" v-model="brand.status"
value="0">禁用
      <input type="radio" name="status" v-model="brand.status"
value="1">启用<br>

      <input type="button" id="btn" @click="submitForm" value="提交">
  </form>
</div>
```

整体页面代码如下:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>添加品牌</title>
</head>
<body>
<div id="app">
  <h3>添加品牌</h3>
  <form action="" method="post">
    品牌名称: <input id="brandName" v-model="brand.brandName"
name="brandName"><br>
    企业名称: <input id="companyName" v-model="brand.companyName"
name="companyName"><br>
    排序: <input id="ordered" v-model="brand.ordered" name="ordered"><br>
    描述信息: <textarea rows="5" cols="20" id="description" v-
model="brand.description" name="description"></textarea><br>
    状态:
      <input type="radio" name="status" v-model="brand.status" value="0">禁用
      <input type="radio" name="status" v-model="brand.status" value="1">启用
<br>

      <input type="button" id="btn" @click="submitForm" value="提交">
  </form>
</div>
<script src="js/axios-0.18.0.js"></script>
```

```

<script src="js/vue.js"></script>
<script>
  new Vue({
    el: "#app",
    data(){
      return {
        brand:{}
      }
    },
    methods:{
      submitForm(){
        // 发送ajax请求, 添加
        var _this = this;
        axios({
          method:"post",
          url:"http://localhost:8080/brand-demo/addServlet",
          data:_this.brand
        }).then(function (resp) {
          // 判断响应数据是否为 success
          if(resp.data == "success"){
            location.href = "http://localhost:8080/brand-
demo/brand.html";
          }
        })
      }
    }
  })
</script>
</body>
</html>

```

通过上面的优化，前端代码确实简化了不少。但是页面依旧是不怎么好看，那么接下来我们学习 Element，它可以美化页面。