

案例

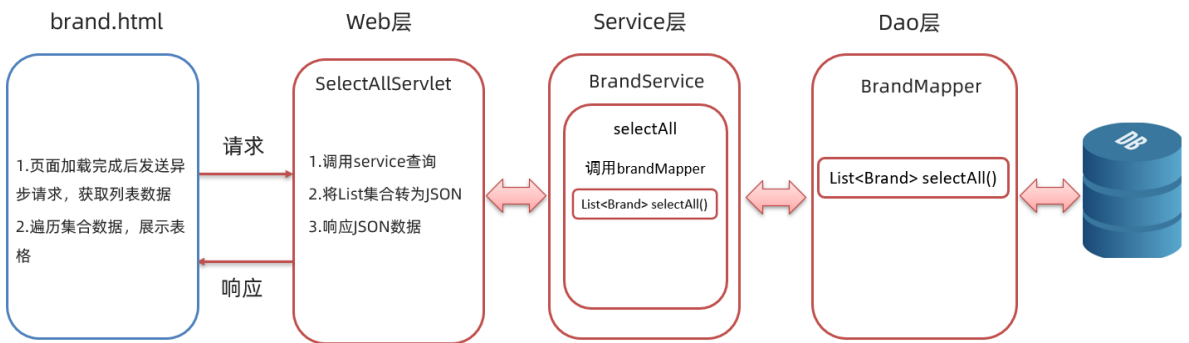
需求

使用Axios + JSON 完成品牌列表数据查询和添加。页面效果还是下图所示：

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

查询所有功能

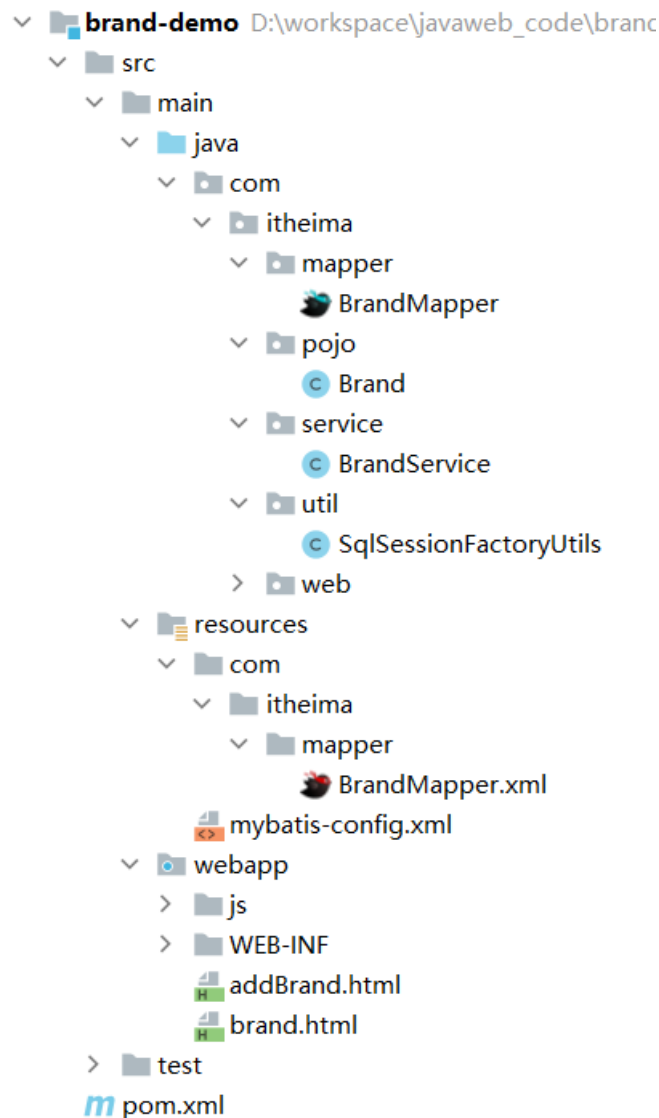


如上图所示就该功能的整体流程。前后端需以JSON格式进行数据的传递；由于此功能是查询所有的功能，前端发送ajax请求不需要携带参数，而后端响应数据需以如下格式的json数据

```
[{"brandName": "三只松鼠", "companyName": "三只松鼠股份有限公司", "description": "好吃不上火", "id": 1, "ordered": 5, "status": 0, "statusStr": "禁用"}, {"brandName": "华为", "companyName": "华为技术有限公司", "description": "华为致力于把数字世界带入每个人、每个家庭、每个组织，构建万物互联的智能世界", "id": 2, "ordered": 100, "status": 1, "statusStr": "启用"}, {"brandName": "小米", "companyName": "小米科技有限公司", "description": "are you ok", "id": 3, "ordered": 50, "status": 1, "statusStr": "启用"}, {"brandName": "鸿星尔克", "companyName": "鸿星尔克", "description": "to be no.1", "id": 6, "ordered": 10, "status": 0, "statusStr": "禁用"}, {"brandName": "黑马", "companyName": "黑马", "description": "发送到发大水", "id": 7, "ordered": 100, "status": 1, "statusStr": "启用"}]
```

环境准备

将 02-AJAX\04-资料\3. 品牌列表案例\初始工程 下的 brand-demo 工程拷贝到我们自己 工作空间 ，然后再将项目导入到我们自己的 Idea 中。工程目录结构如下：



==注意: ==

- 在给定的原始工程中已经给定一些代码。而在此案例中我们只关注前后端交互代码实现
- 要根据自己的数据库环境去修改连接数据库的信息，在 `mybatis-config.xml` 核心配置文件中修改

后端实现

在 `com.itheima.web` 包下创建名为 `SelectAllServlet` 的 `servlet`，具体的逻辑如下：

- 调用 service 的 `selectAll()` 方法进行查询所有的逻辑处理
- 将查询到的集合数据转换为 json 数据。我们将此过程称为 ==序列化==；如果是将 json 数据转换为 Java 对象，我们称之为 ==反序列化==
- 将 json 数据响应回给浏览器。这里一定要设置响应数据的类型及字符集
`response.setContentType("text/json;charset=utf-8");`

`SelectAllServlet` 代码如下：

```
@WebServlet("/selectAllServlet")
public class SelectAllServlet extends HttpServlet {
    private BrandService brandService = new BrandService();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //1. 调用Service查询
```

```

        List<Brand> brands = brandService.selectAll();

        //2. 将集合转换为JSON数据    序列化
        String jsonString = JSON.toJSONString(brands);

        //3. 响应数据    application/json    text/json
        response.setContentType("text/json;charset=utf-8");
        response.getWriter().write(jsonString);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        this.doGet(request, response);
    }
}

```

前端实现

1. 引入 js 文件

在 `brand.html` 页面引入 `axios` 的 js 文件

```
<script src="js/axios-0.18.0.js"></script>
```

2. 绑定 页面加载完毕 事件

在 `brand.html` 页面绑定加载完毕事件，该事件是在页面加载完毕后被触发，代码如下

```

window.onload = function() {

}

```

3. 发送异步请求

在页面加载完毕事件绑定的匿名函数中发送异步请求，代码如下：

```

//2. 发送ajax请求
axios({
    method:"get",
    url:"http://localhost:8080/brand-demo/selectAllServlet"
}).then(function (resp) {

});

```

4. 处理响应数据

在 `then` 中的回调函数中通过 `resp.data` 可以获取响应回来的数据，而数据格式如下

```
[{
  "brandName": "三只松鼠",
  "companyName": "三只松鼠股份有限公司",
  "description": "好吃不上火",
  "id": 1,
  "ordered": 5,
  "status": 0,
  "statusStr": "禁用"
}, {
  "brandName": "华为",
  "companyName": "华为技术有限公司",
  "description": "华为致力于把数字世界带入每个人、每个家庭、每个组织，构建万物互联的智能世界",
  "id": 2,
  "ordered": 100,
  "status": 1,
  "statusStr": "启用"
}, {
```

现在我们需要拼接字符串，将下面表格中的所有的 `tr` 拼接到一个字符串中，然后使用

`document.getElementById("brandTable").innerHTML = 拼接好的字符串` 就可以动态的展示出用户想看到的数据

```
<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>
  <tr align="center">
    <td>1</td>
    <td>三只松鼠</td>
    <td>三只松鼠</td>
    <td>100</td>
    <td>三只松鼠，好吃不上火</td>
    <td>启用</td>
    <td><a href="#">修改</a> <a href="#">删除</a></td>
  </tr>
```

而表头行是固定的，所以先定义初始值是表头行数据的字符串，如下

```
//获取数据
let brands = resp.data;
let tableData = " <tr>\n" +
  "      <th>序号</th>\n" +
  "      <th>品牌名称</th>\n" +
  "      <th>企业名称</th>\n" +
  "      <th>排序</th>\n" +
  "      <th>品牌介绍</th>\n" +
  "      <th>状态</th>\n" +
  "      <th>操作</th>\n" +
  "    </tr>";
```

接下来遍历响应回来的数据 `brands`，拿到每一条品牌数据

```

for (let i = 0; i < brands.length ; i++) {
    let brand = brands[i];
}

```

紧接着就是从 brand 对象中获取数据并且拼接 数据行，累加到 tableData 字符串变量中

```

tableData += "\n" +
    "    <tr align=\"center\">\n" +
    "        <td>"+(i+1)+"</td>\n" +
    "        <td>"+brand.brandName+"</td>\n" +
    "        <td>"+brand.companyName+"</td>\n" +
    "        <td>"+brand.ordered+"</td>\n" +
    "        <td>"+brand.description+"</td>\n" +
    "        <td>"+brand.status+"</td>\n" +
    "\n" +
    "        <td><a href=\"#\">修改</a> <a href=\"#\">删除</a></td>\n" +
    "    </tr>";

```

最后再将拼接好的字符串写到表格中

```

// 设置表格数据
document.getElementById("brandTable").innerHTML = tableData;

```

整体页面代码如下：

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<a href="addBrand.html"><input type="button" value="新增"></a><br>
<hr>
<table id="brandTable" border="1" cellpadding="0" width="100%">

</table>

<script src="js/axios-0.18.0.js"></script>

<script>
    //1. 当页面加载完成后，发送ajax请求
    window.onload = function () {
        //2. 发送ajax请求
        axios({
            method:"get",
            url:"http://localhost:8080/brand-demo/selectAllServlet"
        }).then(function (resp) {
            //获取数据
            let brands = resp.data;
            let tableData = " <tr>\n" +
                "        <th>序号</th>\n" +
                "        <th>品牌名称</th>\n" +
                "        <th>企业名称</th>\n" +

```

```

"        <th>排序</th>\n" +
"        <th>品牌介绍</th>\n" +
"        <th>状态</th>\n" +
"        <th>操作</th>\n" +
"    </tr>";

for (let i = 0; i < brands.length ; i++) {
    let brand = brands[i];

    tableData += "\n" +
        "    <tr align=\"center\">\n" +
        "        <td>"+(i+1)+"</td>\n" +
        "        <td>"+brand.brandName+"</td>\n" +
        "        <td>"+brand.companyName+"</td>\n" +
        "        <td>"+brand.ordered+"</td>\n" +
        "        <td>"+brand.description+"</td>\n" +
        "        <td>"+brand.status+"</td>\n" +
        "\n" +
        "        <td><a href=\"#\">修改</a> <a href=\"#\">删除</a>
    </td>\n" +
        "    </tr>";
}
// 设置表格数据
document.getElementById("brandTable").innerHTML = tableData;
})
}
</script>
</body>
</html>

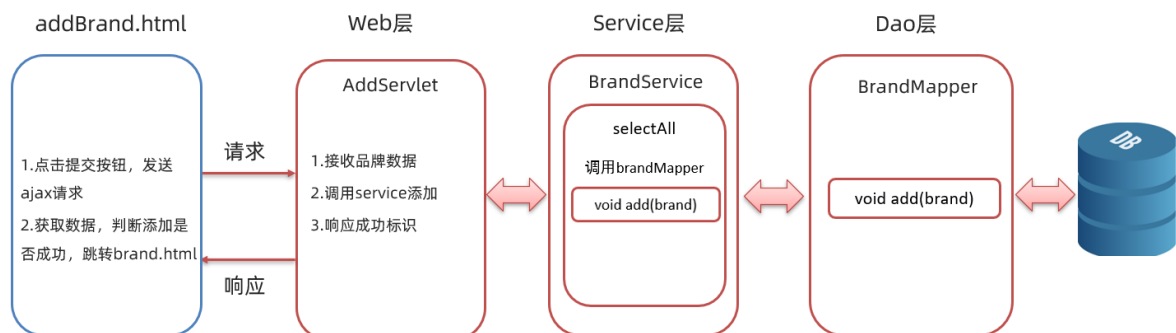
```

添加品牌功能



如上所示, 当我们点击 **新增** 按钮, 会跳转到 `addBrand.html` 页面。在 `addBrand.html` 页面输入数据后点击 **提交** 按钮, 就会将数据提交到后端, 而后端将数据保存到数据库中。

具体的前后端交互的流程如下:



==说明: ==

前端需要将用户输入的数据提交到后端, 这部分数据需要以 json 格式进行提交, 数据格式如下:

```
{
  "brandName": "鸿星尔克",
  "companyName": "鸿星尔克",
  "ordered": "200",
  "description": "to be no.1",
  "status": "1"
}
```

后端实现

在 `com.itheima.web` 包下创建名为 `AddServlet` 的 `servlet`，具体的逻辑如下：

- 获取请求参数

由于前端提交的是 json 格式的数据，所以我们不能使用 `request.getParameter()` 方法获取请求参数

- 如果提交的数据格式是 `username=zhangsan&age=23`，后端就可以使用 `request.getParameter()` 方法获取
 - 如果提交的数据格式是 json，后端就需要通过 `request` 对象获取输入流，再通过输入流读取数据
- 将获取到的请求参数（json格式的数据）转换为 `Brand` 对象
 - 调用 `service` 的 `add()` 方法进行添加数据的逻辑处理
 - 将 json 数据响应回给浏览器。

`AddServlet` 代码如下：

```
@WebServlet("/addServlet")
public class AddServlet extends HttpServlet {

    private BrandService brandService = new BrandService();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        //1. 接收数据,request.getParameter 不能接收json的数据
        /* String brandName = request.getParameter("brandName");
        System.out.println(brandName);*/

        // 获取请求体数据
        BufferedReader br = request.getReader();
        String params = br.readLine();
        // 将JSON字符串转为Java对象
        Brand brand = JSON.parseObject(params, Brand.class);
        //2. 调用service 添加
        brandService.add(brand);
        //3. 响应成功标识
        response.getWriter().write("success");
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

前端实现

在 `addBrand.html` 页面给 `提交` 按钮绑定点击事件，并在绑定的匿名函数中发送异步请求，代码如下：

```
//1. 给按钮绑定单击事件
```

```
document.getElementById("btn").onclick = function () {
    //2. 发送ajax请求
    axios({
        method:"post",
        url:"http://localhost:8080/brand-demo/addServlet",
        data:???
    }).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";
        }
    })
}
```

现在我们只需要考虑如何获取页面上用户输入的数据即可。

首先我们先定义如下的一个 js 对象，该对象是用来封装页面上输入的数据，并将该对象作为上面发送异步请求时 `data` 属性的值。

```
// 将表单数据转为json
var formData = {
    brandName:"",
    companyName:"",
    ordered:"",
    description:"",
    status:"",
};
```

接下来获取输入框输入的数据，并将获取到的数据赋值给 `formData` 对象指定的属性。比如获取用户名的输入框数据，并把该数据赋值给 `formData` 对象的 `brandName` 属性

```
// 获取表单数据
let brandName = document.getElementById("brandName").value;
// 设置数据
formData.brandName = brandName;
```

==说明：其他的输入框都用同样的方式获取并赋值。==但是有一个比较特殊，就是状态数据，如下图是页面内容

状态：

```
<input type="radio" name="status" value="0">禁用
<input type="radio" name="status" value="1">启用<br>
```

我们需要判断哪儿个被选中，再将选中的单选框数据赋值给 `formData` 对象的 `status` 属性，代码实现如下：

```
let status = document.getElementsByName("status");
for (let i = 0; i < status.length; i++) {
    if(status[i].checked){
        //
        formData.status = status[i].value ;
    }
}
```

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

  <input type="button" id="btn" value="提交">
</form>

<script src="js/axios-0.18.0.js"></script>

<script>
  //1. 给按钮绑定单击事件
  document.getElementById("btn").onclick = function () {
    // 将表单数据转为json
    var formData = {
      brandName:"",
      companyName:"",
      ordered:"",
      description:"",
      status:"",
    };
    // 获取表单数据
    let brandName = document.getElementById("brandName").value;
    // 设置数据
    formData.brandName = brandName;

    // 获取表单数据
    let companyName = document.getElementById("companyName").value;
    // 设置数据
    formData.companyName = companyName;

    // 获取表单数据
    let ordered = document.getElementById("ordered").value;
    // 设置数据
    formData.ordered = ordered;

    // 获取表单数据
    let description = document.getElementById("description").value;
    // 设置数据
    formData.description = description;

    let status = document.getElementsByName("status");
    for (let i = 0; i < status.length; i++) {
```

```

        if(status[i].checked){
            //
            formData.status = status[i].value ;
        }
    }
    //console.log(formData);
    //2. 发送ajax请求
    axios({
        method:"post",
        url:"http://localhost:8080/brand-demo/addServlet",
        data:formData
    }).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>

```

==说明：==

查询所有 功能和 添加品牌 功能就全部实现，大家肯定会感觉前端的代码很复杂；而这只是暂时的，后面学习了 vue 前端框架后，这部分前端代码就可以进行很大程度的简化。