### vue初始化一个项目

  vue create 项目名【element-ui-form】  
   cd 项目名【element-ui-form】  
   npm run serve 运行

### 安装element-ui框架

  npm install --save element-ui

### 在main.js里面引入element-ui和

   // 引入两个element-ui文件 ElementUI和index.css  
   import ElementUI from 'element-ui'  
   import 'element-ui/lib/theme-chalk/index.css'  
​  
   // 全局使用ElementUI  
   Vue.use(ElementUI)

### 删除home.vue/about.vue或者修改roter里面的index.js[about和home]

  删除router文件夹下的home.vue和about.vue

### 在router文件夹里添加新的内容【index.vue和new.vue】商品列表和新增列表

{//商品列表  
   path: '/',  
   name: 'Products',  
   component: () => import('../views/form/index.vue')  
},  
{//商品新增  
   path: '/new',  
   name: 'New',  
   component: () => import('../views/form/new.vue')  
},  
{// 商品编辑  
   path: '/edit/:id',  
   name: 'Edit',  
   component: () => import('../views/form/edit.vue')  
}

### 商品列表页面 index.vue

在element-ui 输入table 选择自己需要的组件 引入

**html**

   <template>  
   <div class="products">  
     <!-- <h1><router-link to="/new">商品列表首页</router-link></h1> -->  
     <!-- <h1><router-link to="/edit">商品编辑</router-link></h1> -->  
   <!-- 商品查询 -->  
​  
   <!-- 商品列表 -->  
   <el-table :data="products" border style="width: 100%">  
       <!-- 设置六个栏 序号、名字（name）、图片(scope.row.pic)、价格(price)、库存(amount)、操作-->  
     <el-table-column type="index" label="序号" width="50" align="center"></el-table-column>  
     <!-- 通过props传参 绑定name属性 从而获取新增列表页面的name参数 即获得新增列表页面的商品信息 -->  
     <el-table-column type="text" label="名字" prop="name" width="180" align="center"></el-table-column>  
     <el-table-column type="text" label="图片" width="200" align="center">  
       <!-- 插槽 单独修饰图片的大小 -->  
       <template slot-scope="scope">  
         <img :src="scope.row.pic" style="width: 120px; max-height: 200px;" />  
       </template>  
     </el-table-column>  
     <el-table-column type="text" label="价格" prop="price" align="center"></el-table-column>  
     <el-table-column type="text" label="库存" prop="amount" align="center"></el-table-column>  
     <el-table-column type="index" label=操作 width="120" align="center">  
       <template slot-scope="scope">  
         <!-- 编辑[editClickj]和删除[delClick] 使用插槽 -->  
         <el-button @click="editClickj(scope.row)" size="small" type="text">编辑</el-button>  
         <el-button @click="delClick(scope.row)" size="small" type="text">删除</el-button>  
       </template>  
     </el-table-column>  
   </el-table>  
   <!-- 新增按钮 点击事件 -->  
     <el-button @click="addHandle" style="margin-top:20px">商品新增</el-button>  
   </div>  
 </template>

**js**

       <script>  
    export default {  
      name: 'Products',  
      // 商品数据  
      data() {  
        return{  
          products: [], // 获取新增页面的商品数据 在商品列表页面显示  
          // query: ''  
        };  
      },  
      // 渲染数据到页面上  
      created() {  
        // localStorage 存储数据  
        //此处再转  
        try {//获取存入商品到本地{从商品新增页面(new.vue)获取}  
          if (localStorage.getItem('products')) {  
            this.products = JSON.parse(localStorage.getItem('products'));  
          }  
          // 判断是否存入  
        } catch (err){  
          console.log(err)  
        }  
      },  
          methods: {  
            // 点击新增页面 跳转新增页面(new.vue)  
          //点击新增 跳转新增页面（new.vue）  
        addHandle() {  
          this.$router.push({  
            name: "New"  
          });  
        }  
      }  
    }  
     </script>

### 商品新增 new.vue

在element-ui里引入form组件

**html**

   <template>  
     <div class="new">  
       <!-- 商品新增 -->  
     <el-form :model="ruleForm"  :rules="rules" ref="ruleForm" label-width="100px" class="demo-ruleForm">  
     <!-- 五个小表单 名称（name）、图片（pic）、价格（price）、库存（amount）、操作按钮（保存和重置） -->  
     <el-form-item label="名称" prop="name">  
       <el-input v-model="ruleForm.name"></el-input>  
     </el-form-item>  
     <el-form-item label="图片" prop="pic">  
       <el-input v-model="ruleForm.pic"></el-input>  
     </el-form-item>  
     <el-form-item label="价格" prop="price">  
       <el-input v-model="ruleForm.price"></el-input>  
     </el-form-item>  
     <el-form-item label="库存" prop="amount">  
       <el-input v-model="ruleForm.amount"></el-input>  
     </el-form-item>  
     <el-form-item>  
       <!-- 保存和重置按钮 -->  
       <el-button type="primary" @click="submitForm('ruleForm')">保存</el-button>  
       <el-button @click="resetForm('ruleForm')">重置</el-button>  
     </el-form-item>  
   </el-form>  
     </div>  
   </template>

**js**

   <script>  
​  
​  
   export default {  
     name: 'New',  
     // 将表单里的数据反映出来  
     data() {  
       return {  
           list: [],  
           ruleForm: {  
             name: '',  
             pic: '',  
             price: 0,  
             amount: 0  
          },  
           rules: {  
             name: [  
              { required: true, message: '请输入活动名称', trigger: 'blur' },  
              { min: 3, max: 20, message: '长度在 3 到 5 个字符', trigger: 'blur' }  
            ],  
             pic: [  
              { required: true, message: '请输图片地址', trigger: 'blur' },  
              { min: 30, message: '长度不低于20个字符', trigger: 'blur' }  
            ],  
             price: [  
              { required: true, message: '请输入', trigger: 'blur' },  
              { min: 1, max: 10, message: '长度在 0 到 10 个字符', trigger: 'blur' }  
            ],  
             amount: [  
              { required: true, message: '请输入活动名称', trigger: 'blur' },  
              { min: 1, max: 10000, message: '长度在 0 到 10000 个字符', trigger: 'blur' }  
            ]  
          }  
        };  
    },  
     created() {  
       // 初始化数据  
       this.list = [];  
       try{  
         if(localStorage.getItem('products')){  
           this.list = JSON.parse(localStorage.getItem('products'))  
        }  
      } catch (err){  
         console.log(err)  
      }  
    },  
     methods: {  
       //保存  
       submitForm(formName) {  
           this.$refs[formName].validate((valid) => {  
             if (valid) {  
               // alert('submit!');  
               // 原有数据this.list  
               this.list.push({  
                 // 解析一下所有的数据  
                ...this.ruleForm,  
                 // 传入当前商品id  
                 id: Date.now()  
              })  
               // 通过id 将数据存入本地 通过列表页面获取  
                 localStorage.setItem("products",JSON.stringify(this.list));  
                 this.$router.push({//将数据传给列表页面  
                   name: 'Products'  
                });  
            } else {  
               console.log('error submit!!');  
               return false;  
            }  
          });  
        },  
         // 重置  
         resetForm(formName) {  
           this.$refs[formName].resetFields();  
        }  
    }  
  }