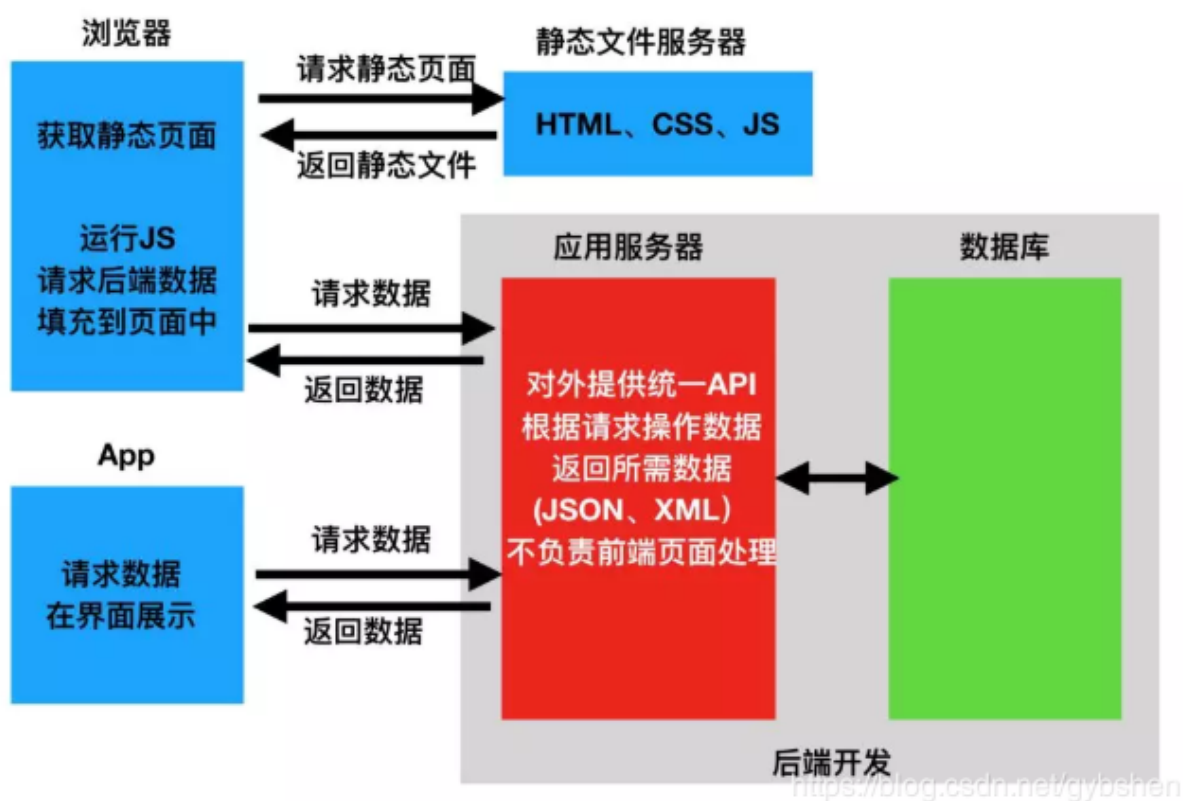


## 课程介绍

今天我们将完成一个 客户信息展示 的示例，这个示例使用 前后端分离的开发模式来完成。

以下是前后端分离的开发模式框架示意图：



这个示例的效果以下：

新增

删除

选择	编号	客户名	联系电话	地址	...
<input type="checkbox"/>	1				
<input checked="" type="checkbox"/>	2				
<input type="checkbox"/>	3				
<input checked="" type="checkbox"/>	4				
<input checked="" type="checkbox"/>	5				

1

2

3

弹出窗口

客户名称

性别

☒男☐女

联系电话

所在地

保存

取消

要达到以上的效果，只需要10个步骤就可以做出来了。

课程目录：

课程介绍

以下是前后端分离的开发模式框架示意图：

这个示例的效果以下：

第1节：建立数据库表

第2节：创建一个后端项目

- 1)、创建一个SpringBoot项目
- 2)、引入所需依赖jar包
- 3)、配置数据源连接

第3节：编写API接口程序

第4节：配置接口映射和分页插件的注册

第5节：启动后台项目，测试API接口程序

## 第6节：创建一个前端项目，并新建页面

- 1)、基础的准备
- 2)、模拟列表数据

## 第7节：绑定后台列表数据

## 第8节：列表分页设置

## 第9节：删除功能

## 第10节：新建、编辑功能

作者介绍：

## 开发环境和相关工具：

前端：Element UI + VUE

后端：SpringBoot、MybatisPlus、Lombok、EasyCode代码生成器

后端开发工具：IDEA

API测试工具：PostMan

前端开发工具：VSCode

学习目的：了解前端、后端、相关工具使用，以及完整的项目实例，帮助新手快速入门。

# 第1节：建立数据库表

mysql,

customer\_db

customer\_table

# 第2节：创建一个后端项目

工具：IDEA。用快速创建方式创建一个SpringBoot项目

## 1)、创建一个SpringBoot项目

## 2)、引入所需依赖jar包

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.1.9.RELEASE</version>
```

```

        <relativePath/> <!-- lookup parent from repository -->
    </parent>

<dependencies>
    <!-- spring boot 启动器-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <!-- spring boot web 启动器-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- spring boot test 启动器-->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <!-- lombok简化java代码 如果没有安装, 先安装这个插件-->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>

    <!-- mybatis-plus插件 -->
    <dependency>
        <groupId>com.baomidou</groupId>
        <artifactId>mybatis-plus-boot-starter</artifactId>
        <version>3.1.0</version>
    </dependency>

    <!-- mysql jdbc驱动 -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>

```

```
        <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
</plugins>
</build>
```

如果用maven创建项目，需要手工创建程序启动类：

```
@SpringBootApplication
public static void main(String[] args) {
    SpringApplication.run (xx.class, args);
}
```

如果用maven创建项目，默认是没有测试类的，如果需要创建测试类，需要在测试类头增加注解：

```
@RunWith(SpringRunner.class)
@SpringBootTest
```

### 3)、配置数据源连接

新建application.yml配置文件

```
server:
  port: 8088
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3309/XXXX
    username: root
    password: 8888
mybatis-plus:
  #信息输出设置
  configuration:
    log-impl: org.apache.ibatis.logging.stdout.StdoutImpl
```

## 第3节：编写API接口程序

为了高效，使用EasyCode工具来生成后台java代码，也就是API接口程序。

## 第4节：配置接口映射和分页插件的注册

- 1)、在主程序添加注解 @MapperScan，作用是：一次性扫描Mapper映射接口的包路径；
- 2)、在项目中加入配置类，在类中用@Bean注解来注册分页插件；

```
//Spring boot方式
@EnableTransactionManagement
@Configuration
public class MybatisPlusConfig {
    @Bean
    public PaginationInterceptor paginationInterceptor() {
        PaginationInterceptor paginationInterceptor = new
        PaginationInterceptor();
        // 设置请求的页面大于最大页后操作， true调回到首页， false 继续请求 默认false
        // paginationInterceptor.setOverflow(false);
        // 设置最大单页限制数量，默认 500 条，-1 不受限制
        // paginationInterceptor.setLimit(500);
        return paginationInterceptor;
    }
}
```

## 第5节：启动后台项目，测试API接口程序

开启 PostMan工具测试。get、post

## 第6节：创建一个前端项目，并新建页面

前端用：Element UI+VUE。手工简单创建项目（不使用vue-cli）

用vsCode工具编辑

### 1)、基础的准备

### 2)、模拟列表数据

## 第7节：绑定后台列表数据

vue基本方法框架：

```
<script>
```

```

var Main = {
  data() {
    return {

    }
  },

  //web页面初始化时, 调用
  created() {

  },

  //定义一些方法, 放在这里面
  methods: {

  }
}
var Ctor = Vue.extend(Main)
new Ctor().$mount('#app')
</script>

```

```

// 用ajax去请求服务端数据
getData() {
  let that = this;
  let url = "http://localhost:8089/customerTable";
  that.$http.get(url, null, { emulateJSON: true })
    .then(res => {
      let rs = JSON.parse(res.body);
      if (rs.code == 0) {
        that.tableData = rs.data.records;
      }
    });
}

```

获取远程数据:

```

this.$http.get(url, data, {emulateJSON:true})
  .then( res => {
    let rs = JSON.parse(res.body);
  });

```

封装好的ajax:

```

/**
 * 简单封装了ajax对象

```

```

*/
class Ajax {

    vueObj = null;

    baseUrl = "http://localhost:8099/";

    /**
     * 构造方法
     * @param {*} vueObj
     */
    constructor(vueObj) {

        this.vueObj = vueObj;
    }

    /**
     * get方法
     * @param {*} api
     * @param {*} callback
     * @param {*} params
     */
    get(api, callback, params) {
        let url = this.baseUrl + api ;
        let data = JSON.stringify(params);
        this.vueObj.$http.get(url, data, {emulateJSON:true})
            .then( res => {
                callback(JSON.parse(res.body));
            });
    }

    /**
     * post方法
     * @param {*} api
     * @param {*} callback
     * @param {*} params
     */
    post(api, callback, params) {
        let url = this.baseUrl + api ;
        let data = JSON.stringify(params);
        this.vueObj.$http.post(url, data, {emulateJSON:true})
            .then( res => {
                callback(JSON.parse(res.body));
            });
    }

    /**
     * put方法
     * @param {*} api

```



```

    * @param {*} callback
    * @param {*} params
    */
    put(api,callback,params){
        let url = this.baseUrl + api ;
        let data = JSON.stringify(params);
        this.vueObj.$http.put(url, data,{emulateJSON:true})
        .then( res => {
            callback(JSON.parse(res.body));
        });
    }

    /**
     * delete方法
     * @param {*} api
     * @param {*} callback
     * @param {*} params
     */
    delete(api,callback,params){
        let url = this.baseUrl + api ;
        let data = JSON.stringify(params);
        this.vueObj.$http.delete(url, data ,{emulateJSON:true})
        .then( res => {
            callback(JSON.parse(res.body));
        });
    }
}

```

## 第8节：列表分页设置

主要代码：

```

//分页点击事件
currentChange(current){
    // console.log(current);
    this.page.current = current;
    this.getData();
}

```

```
<el-pagination
  v-if="page.total>page.size"
  background
  layout="prev, pager, next"
  @current-change="currentChange"
  :page-size="page.size"
  :current-page="page.current"
  :total="page.total">
</el-pagination>
```

## 第9节：删除功能

---

```
// 删除数据
del() {
  if (this.idList.length > 0) {
    let that = this;
    let idList = that.idList.toString();
    let url = "customerTable?idList=" + idList;
    //调用Ajax的delete方法
    that.ajax.delete(url, function (rs) {
      if (rs.code == 0) {
        //重新获取数据
        that.getData();
      }
    });
  }
}
```

## 第10节：新建、编辑功能

---

```

onSubmit() {
    let that = this;
    let url = "customerTable";
    let data = that.form;
    //调用Ajax的post方法
    that.ajax.post(url, function (rs) {
        if (rs.code == 0) {
            //重新获取数据
            that.getData();
        }
    },data);
}

```

最终效果：

<input type="checkbox"/>	编号	姓名	性别	联系方式	地址
<input type="checkbox"/>	1	刘德华	男	18816788767	香港
<input type="checkbox"/>	2	张学友	男	18816788767	香港
<input type="checkbox"/>	4	苏有朋	男	18816788767	香港
<input type="checkbox"/>	9	张三	男	19934444	重庆渝北
<input type="checkbox"/>	10	李四	男	18817879877	重庆渝中区

2

回

<input type="checkbox"/>	编号	姓名	性别	联系方式	地址
<input type="checkbox"/>	1	刘德华	男	18816788767	香港
<input type="checkbox"/>	2	张学友			
<input type="checkbox"/>	4	苏有朋			
<input type="checkbox"/>	9	张三			
<input type="checkbox"/>	10	李四			

<

1

2

>

新增数据

客户名称

性别

☒ 男

☐ 女

联系电话

所在地址

保存


取消

## 作者介绍：


本人真名叫莫小明（小明哥），从业近15年。在IT路上，产品、项目采坑无数。参与过多个大型项目、产品的研发和管理，加微信有更多项目案例分享和技术交流。




无数个项目积累：07~19年：




07-09年




2010年




2011年




2012年




2013年




华南地区：14-17




2016




2017




2018




2019




典型案例




www



ProjectWork



优秀产品



项目进度-模板.xlsx

2013年



0-公司产品



门户改造项目



1.1-门户改造



2-1厂项目



3-J集团BPM项目



6-江电力



7-河南



8-集团综合



9-



10-



11-医院B



12-儿童医院



13-质量管理



14-集团综合业务信...二期项目



15-物流综合信息平台项目



集成数据中心项目需求



portal需求



集团数据集成



威孚集团



物联网项目



信托



项目验收资料



管理报表项目



木本地化维护



5-东



机场项目



专家私项目



4-佛项目



场



交流资料-8-21

