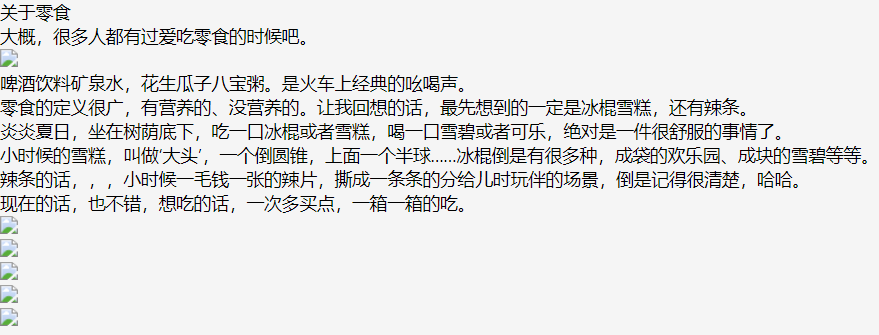
# 前端

## 基础

三件套：HTML、CSS、JavaScript。[B站尚硅谷前端课程链接](https://www.bilibili.com/video/BV1XJ411X7Ud)

### HTML

<div id="lingshi">  
 <h2>关于零食</h2>  
 <p class="sketch">大概，很多人都有过爱吃零食的时候吧。</p>  
 <div class="sentence">  
 <img src="../pic/eat/snacks.jpg" >  
 <p>啤酒饮料矿泉水，花生瓜子八宝粥。是火车上经典的吆喝声。</p>  
 <p>零食的定义很广，有营养的、没营养的。让我回想的话，最先想到的一定是冰棍雪糕，还有辣条。</p>  
 <p>炎炎夏日，坐在树荫底下，吃一口冰棍或者雪糕，喝一口雪碧或者可乐，绝对是一件很舒服的事情了。</p>  
 <p>小时候的雪糕，叫做‘大头’，一个倒圆锥，上面一个半球……冰棍倒是有很多种，成袋的欢乐园、成块的雪碧等等。</p>  
 <p>辣条的话，，，小时候一毛钱一张的辣片，撕成一条条的分给儿时玩伴的场景，倒是记得很清楚，哈哈。</p>  
 <p>现在的话，也不错，想吃的话，一次多买点，一箱一箱的吃。</p>  
 </div>  
 <div class="imgBox">  
 <ul>  
 <li><img src="../pic/eat/icecream.jpg" ></li>  
 <li><img src="../pic/eat/spicyslice.jpg" ></li>  
 <li><img src="../pic/eat/nut.jpg" ></li>  
 <li><img src="../pic/eat/jelly.jpg" ></li>  
 <li><img src="../pic/eat/Chocolates.jpg" ></li>  
 </ul>  
 </div>  
</div>



HTML就是标签语言，不同的标签有不同的含义，用于展示不同的元素。

### CSS

/\* 三块内容 \*/  
#eatTitle,#zhushi,#lingshi{  
 /\* 设置三个div的上右下左外边距 \*/  
 margin: 10% 8% 15% 8%;  
}  
  
#eatTitle h2{  
 /\* 设置title的h2,竖排显示（宽度和字体大小一样) \*/  
 margin-right: 50px;  
 float: left;  
 width: 30px;  
 font-size: 30px;  
}  
/\* 每一部分sketch类p标签上下边距 \*/  
.sketch{  
 margin-top: 20px;  
 margin-bottom: 30px;  
}

通过css选择器可以修改选择的元素的样式、添加动画效果，使页面看起来更美观。

### JavaScript

function appendChildNavs()  
{  
 //四个导航，及其下拉菜单，以及对应url  
 var navs=['-吃-','-喝-','-玩-','-学-'];  
 // 获取div元素  
 var topnav=document.getElementById("topnav");  
 for(var i=0;i<navs.length;i++)  
 {  
 //使用下面的函数生成导航栏，并加入topnav的div中  
 var dropNav=createDropdowndiv(i,navs);  
 topnav.appendChild(dropNav);  
 }  
}

通过JavaScript可以实现与用户的交互，比如给某元素添加点击事件监听器，当用户点击该元素后可以触发该事件（提交表单、请求数据等）。

### 总结

这一部分比较重要的就是CSS布局、元素定位，学完JavaScript基础语法后可以学习一下ES6新特性。

## 进阶

[B站尚硅谷Vue课程链接](https://www.bilibili.com/video/BV1Zy4y1K7SH)

目前流行的前端框架有：[Angular](https://angular.cn/)、[React](https://react.docschina.org/)、[Vue](https://cn.vuejs.org/)。[对比](https://zhuanlan.zhihu.com/p/359540593)

通过使用框架和组件库可以快速开发出一个web网站。

（个人使用过的有jQuery+Lay-UI、Vue+Element-UI，前者稍有过时）

# 后端

（前端转后端可以通过JavaScript入门，下一个Node.js、学习一下express框架就可以了。）

## 基础

### Java基础语法

[B站黑马程序员Java课程链接](https://www.bilibili.com/video/BV1Cv411372m)

Java是编译型和解释型语言。

在学习的过程中需要很好地理解面向对象，熟悉Java中各种集合的特性和方法，了解Stream流、File相关类、Date相关类、IO等。

其中最重要的是代码规范，建议安装阿里代码规范插件Alibaba Java Coding Guidelines。

### JDBC

[B站尚硅谷JDBC课程链接](https://www.bilibili.com/video/BV1eJ411c7rf) （下一模块中的视频链接内有JDBC）。

虽然之后都是使用Mybatis进行数据库操作，但建议还是了解一下这一块。

### Java Web

[B站尚硅谷JavaWeb课程链接](https://www.bilibili.com/video/BV1AS4y177xJ)

这一块比较重要的是对于Web服务相关的概念的理解，例如MVC、Tomcat、Servlet、DispatcherServlet、Session和Cookies等。

## 进阶

目前流行的框架就是Spring Boot了，但是在学习之前，建议学习一下SSM（[B站黑马程序员SSM课程链接](https://www.bilibili.com/video/BV1WZ4y1P7Bp)）：Spring、Spring MVC、Mybatis。Spring是一个轻量级的IOC容器框架，Spring MVC容器是Spring容器的子容器（[详情](https://www.cnblogs.com/ibcdwx/p/13821939.htm)），Mybatis是一个ORM持久层框架。

在学习完SSM之后，上手spring boot就非常简单了，通过引入starter可以省去SSM中的大量配置文件进而实现快速开发。

# 数据库

[B站尚硅谷MySQL课程链接](https://www.bilibili.com/video/BV1iq4y1u7vj)

学习阶段基本使用的都是MySQL，其中5.x和8.x区别较大。[下载链接](https://downloads.mysql.com/archives/community/)

学习中需要重点理解**范式、锁、事务、索引、日志**。

# 应急指挥管理系统

## 环境

1） 数据库及版本：

Mysql5.7.24

2） 后端开发框架及版本：

JDK1.8

底座：SpringBoot2.2.1

数据库操作 Mybatis-Plus3.3.2

3） 前端开发框架、组件及版本：

node:v14.19.0

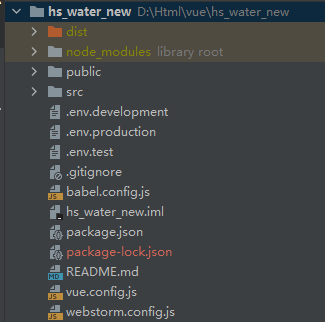
NPM:6.14.16

vue:2.5.2

vue-cli:5.0.4

## 前端

### 目录结构



1. dist：该目录是开发完毕后，经过打包生成的目录（即开发结果），将其中的文件放在web服务器中就可以访问了。
2. node\_modules：依赖文件，在开发过程中导入的各种依赖库就在这里，不用管。
3. public：任何放置在 public 文件夹的静态资源都会被简单的复制，而不经过 webpack。你需要通过绝对路径来引用它们。默认存放index.html
4. **src**：源代码和其他静态资源目录。
5. .env.xxx：多环境配置文件，其中能够以键值对形式声明一些变量，而后可以通过process.env.xxx获取。在后期多环境打包时可以针对不同环境建立不同的配置文件。

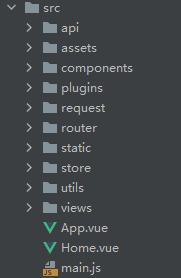
* // 环境变量（这里的环境变量是跟打包有关的，production则会进行压缩代码等，真正跟每个环境有关的变量是下面以VUE\_APP开头的变量）  
  NODE\_ENV=production  
  // 以 VUE\_APP\_ 开头的变量会被 webpack.DefinePlugin 静态嵌入到客户端侧的包中  
  VUE\_APP\_ENV = 'test'  
  VUE\_APP\_Access\_Control\_Allow\_Origin = 'http://localhost:8081'  
  VUE\_APP\_BASE\_API = 'http://localhost:8080'
* NODE\_ENV = 'development'  
  #VUE\_APP\_BASE\_API = 'http://localhost:8080'  
  VUE\_APP\_Access\_Control\_Allow\_Origin = 'http://localhost:8081'  
  VUE\_APP\_BASE\_API = '/api'

1. .gitignore：在使用git提交代码时需要忽略的文件
2. babel.config.js：es6代码的一些转码设置
3. hs\_water\_new.iml：idea针对于该项目的索引文件
4. **package.json**：里面有命令脚本、依赖、开发时依赖等。 在根目录中使用npm run xxx，那么就会执行package.json中scripts中的xxx脚本，比如npm run serve那么就是执行scripts中的serve（vue-cli-service serve --open --hot，open参数表示自动使用浏览器打开开发页面，hot表示修改代码后立马更新到页面）。 而vue-cli-service build后面--mode test表示在打包时使用.env.test配置文件，不指定时使用.env.production配置文件。

* {  
   "name": "hs\_water\_new",  
   "version": "0.1.0",  
   "private": true,  
   "scripts": {  
   "serve": "vue-cli-service serve --open --hot",  
   "build": "vue-cli-service build",  
   "build:test": "vue-cli-service build --mode test",  
   "lint": "vue-cli-service lint"  
   },  
   "dependencies": {  
   "axios": "^0.24.0",  
   "core-js": "^3.19.3",  
   "dayjs": "^1.11.0",  
   "echarts": "^5.2.1",  
   "echarts-wordcloud": "^2.0.0",  
   "element-ui": "^2.11.1",  
   "js2wordcloud": "^1.1.12",  
   "vue": "^2.5.2",  
   "vue-router": "^2.8.1",  
   "vuex": "^3.0.1"  
   },  
   "devDependencies": {  
   "@babel/preset-env": "^7.16.4",  
   "@vue/cli-plugin-babel": "^4.5.15",  
   ...  
   },  
   ...  
  }

1. package-loc.json：依据package.json生成的
2. README.md：使用markdown语法写的项目简介。
3. **vue.config.js**：项目配置，如设置别名、关闭语法检查、配置开发时的反向代理等。
4. webstorm.config.js：因为开发时，WebStorm编辑器默认会找寻项目中的webpack.config.js，一般会在其中配置src目录的别名为@，但是现在创建的项目配置文件改为vue.config.js了，且默认设置src目录的别名为@，但是编辑器不识别，导致在开发过程中无法正确的进行代码跳转，因此新建了该文件，并让编辑器使用该配置文件（其中只配置了src目录别名）。

#### src



main.js：入口文件。在该文件中会引入vue等依赖，然后创建一个Vue组件并绑定至public/index.html中id为app的元素上。

App.vue：因为在main.js中将其绑定至html文件中id为app的元素上了，所以它就相当于html中的那个元素了。

Home.vue：其实应该放在views目录中，该应用的首页将会路由至此。

**views目录**：单页面应用通过路由来决定展示那个组件，会通过路由来展示的组件放在这个目录下，与之对应的是components目录。

**components目录**：在其他组件内部的组件存放在这个目录下（不会通过路由进行加载，而是实例化在其他组件中）。

store目录：里面一个index.js，其中将实例化一个[vuex](https://vuex.vuejs.org/zh/guide/)，用于管理全局状态。

router目录：里面一个index.js，其中将实例化一个Vue-Router，声明路由信息。

request目录：里面一个index.js，其中将实例化一个axios对象，可以设置请求拦截器和响应拦截器。

**api目录**：里面的每个js文件都会导入request，并定义异步请求。

plugins目录：存放一些插件，比如按需引入element-ui、echarts等。

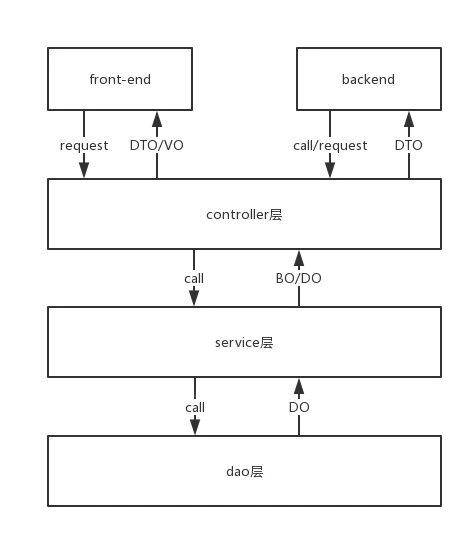
utils目录：存放一些工具，如时间日期工具、节流和防抖等。

assets和static目录：都是存放静态资源，在打包时static目录下的资源不会被webpack处理。[区别](https://www.jianshu.com/p/8341edbc7436)

## 后端

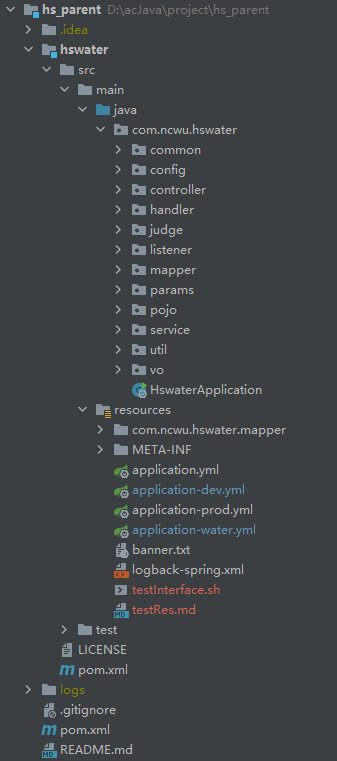
一般web工程结构会分为三层，自下而上是dao层，service层和controller层。

* dao层是数据持久化层，直接进行数据库的读写操作，返回数据对象DO，DO与数据库表一一对应。
* service层为业务层，用来实现业务逻辑。能调用dao层或者service层，返回数据对象DO或者业务对象BO，BO通常由DO转化、整合而来，可以包含多个DO的属性，也可以是只包含一个DO的部分属性。通常为了简便，如果无需转化，service也可以直接返回DO。外部调用（HTTP、RPC）方法也在这一层，对于外部调用来说，service一般会将外部调用返回的DTO转化为BO。
* controller层为控制层，主要处理外部请求。调用service层，将service层返回的BO/DO转化为DTO/VO并封装成统一返回对象返回给调用方。如果返回数据用于前端模版渲染则返回VO，否则一般返回DTO。不论是DTO还是VO，一般都会对BO/DO中的数据进行一些转化和整合，比如将gender属性中的0转化“男”，1转化为“女”等。



### 目录结构

hs\_parent是项目的父模块（其pom.xml定义多环境等），hswater是项目内容。



#### src目录

其中的controller、service、mapper分别对应controller层、service层、dao层。

pojo（Plain Old Java Object）里面就是和数据库表一一对应的类。

vo里面就是返回给前端视图层展示的对象类。

params里面是接收前端请求参数的对象类。

config中是配置信息，如数据源配置、mybatis配置、定时服务配置、线程池配置、WebMVC配置。

common中是全局aop的定义，如日志、缓存等。

handler中定义的是处理器，如全局异常处理。

listener中定义的是监听器，比如监听容器启动后，调用定时服务来调试定时服务。

util中是一些工具类。

judge包中是关于定时研判服务的业务逻辑，之后注入定时任务配置中即可。

#### resource目录

mapper中是与dao层绑定的sql文件。

META-INF就是yml配置文件中的key值声明。

容器默认配置文件是application.yml，通过指定启动项active.profiles可以加载不同的配置文：application-xxx.yml。