

**东软环保公众监督系统**



[**东软环保公众监督系统**](#bookmark1)

[**Part 1 东软环保公众监督系统介绍**](#bookmark2)

[1.1.基本业务流程](#bookmark3)

[1.2.项目技术架构](#bookmark4) [1.2.1.开发环境](#bookmark5) [1.2.2.运行环境](#bookmark6)

[1.3.页面原型](#bookmark7)

[1.3.1.NEPS公众监督员端](#bookmark8) [1.3.2.NEPG网格员端](#bookmark9)

[1.3.3.NEPM系统管理员端](#bookmark10) [1.3.4.NEPV决策者端](#bookmark11)

[1.4.数据库设计](#bookmark12) [1.4.1.ER图](#bookmark13)

[1.4.2.数据库设计总表](#bookmark14) [1.4.3.数据库设计](#bookmark15)

[**Part 2 服务器端接口API**](#bookmark16)

[2.1.admins](#bookmark17)

[2.2.aqi](#bookmark18)

[2.3.aqiFeedback](#bookmark19) [2.4.gridCity](#bookmark20)

[2.5.gridMember](#bookmark21) [2.6.gridProvince](#bookmark22) [2.7.statistics](#bookmark23)

[2.8.supervisor](#bookmark24)

[**Part 3 NEPS公众监督员端APP**](#bookmark25)

[2.1.服务器端项目](#bookmark26) [2.2.前端项目](#bookmark27)

[2.2.1.前端项目搭建](#bookmark28)

[2.2.1.1.开发工具检查](#bookmark29) [2.2.1.2.搭建VueCli工程](#bookmark30)

[2.2.1.3.添加其它依赖及配置文件](#bookmark31) [2.2.1.4.main.js文件](#bookmark32)

[2.2.1.5.App.vue文件](#bookmark33)

[2.2.1.6.路由index.js文件](#bookmark34) [2.2.2.前端代码](#bookmark35)

[**Part 4 NEPG网格员端APP**](#bookmark36)

[2.1.服务器端项目](#bookmark37) [2.2.前端项目](#bookmark38)

[2.2.1.前端项目搭建](#bookmark39)

[2.2.1.1.开发工具检查](#bookmark40) [2.2.1.2.搭建VueCli工程](#bookmark41)

[2.2.1.3.添加其它依赖及配置文件](#bookmark42)

[2.2.1.4.main.js文件](#bookmark43) [2.2.1.5.App.vue文件](#bookmark44)

[2.2.1.6.路由index.js文件](#bookmark45) [2.2.2.前端代码](#bookmark46)

[**Part 5 NEPM系统管理端APP**](#bookmark47)

[2.1.服务器端项目](#bookmark48)

[2.1.1.服务器端项目搭建](#bookmark49)

[2.1.1.1.开发工具检查](#bookmark50)

[2.1.1.2.搭建SpringBoot工程](#bookmark51)

[2.1.2.服务器端代码](#bookmark52) [2.2.前端项目](#bookmark53)

[2.2.1.前端项目搭建](#bookmark54)

[2.2.1.1.开发工具检查](#bookmark55) [2.2.1.2.搭建VueCli工程](#bookmark56)

[2.2.1.3.添加其它依赖及配置文件](#bookmark57)

[2.2.1.4.main.js文件](#bookmark58)

[2.2.1.5.App.vue文件](#bookmark59)

[2.2.1.6.路由index.js文件](#bookmark60)

[2.2.1.7.Vuex的index.js文件](#bookmark61) [2.2.2.前端代码](#bookmark62)

[**Part 6 NEPV决策者端APP**](#bookmark63)

[2.1.服务器端项目](#bookmark64) [2.2.前端项目](#bookmark65)

[2.2.1.前端项目搭建](#bookmark66)

[2.2.1.1.开发工具检查](#bookmark67) [2.2.1.2.搭建VueCli工程](#bookmark68)

[2.2.1.3.添加其它依赖及配置文件](#bookmark69) [2.2.1.4.main.js文件](#bookmark70)

[2.2.1.5.App.vue文件](#bookmark71)

[2.2.1.6.路由index.js文件](#bookmark72) [2.2.2.前端代码](#bookmark73)

**Part 1 东软环保公众监督系统介绍**

|  |
| --- |
| **学习目标**  基本业务流程  项目技术架构  用例图  时序图  类图  页面原型设计  数据库设计 |
| **1.1.基本业务流程** |

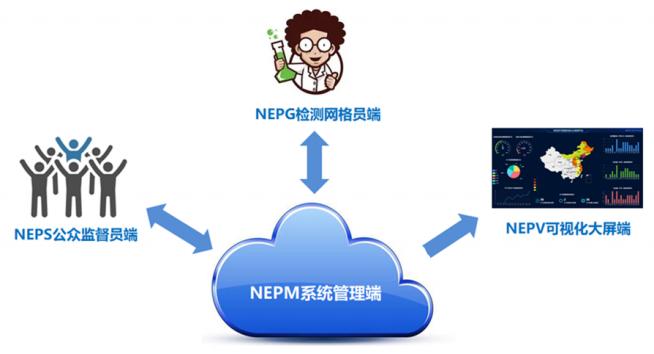
本项目为四端架构：

 NEPS端：公众监督员端。完成公众监督空气质量的全部业务流程。

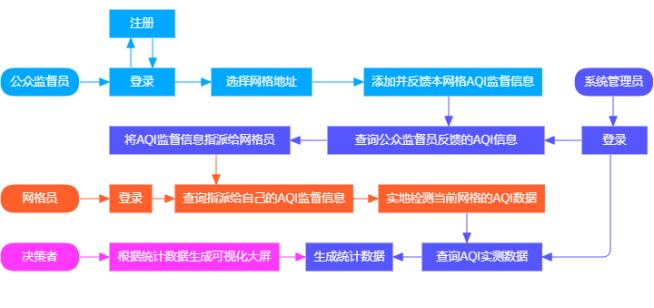
 NEPG端：AQI检测网格员端。完成对公众监督员反馈的空气质量问题做AQI确认的全部业务流程。

 NEPM端：系统管理端。完成对公众监督员反馈的空气质量问题做指派处理，并对AQI检测网格员提交的AQI 数据做统计处理，用于给决策者决策的依据。

 NEPV端：可视化大屏端。将系统管理端统计数据用可视化的方式展示给决策者。



业务流程图：



**1.2.项目技术架构**



**1.2.1.开发环境**



**1.2.2.运行环境**

请添加图片描述

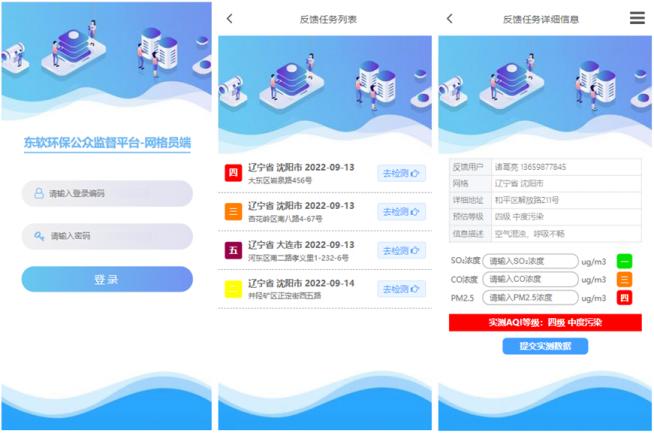
**1.3.页面原型**

**1.3.1.NEPS公众监督员端**





**1.3.2.NEPG网格员端**



**1.3.3.NEPM系统管理员端**





















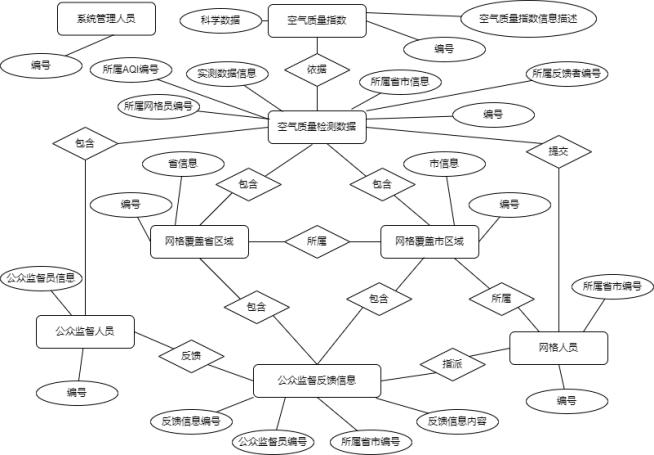
**1.3.4.NEPV决策者端**



**1.4.数据库设计**

**1.4.1.ER图**

东软环保公众监督系统数据库设计-ER图



**1.4.2.数据库设计总表**

东软环保公众监督系统数据库设计书-总表：



**1.4.3.数据库设计**

1. 空气质量指数级别表（aqi）



2. 空气质量监测网格员表（grid\_member）



3. 系统网格覆盖省区域表（grid\_province）



4. 系统网格覆盖市区域表（grid\_city）



5. 空气质量公众监督反馈表（aqi\_feedback）



6. 空气质量监测数据统计表（statistics）



7. 系统管理员表（admins）



8. 公众监督员表（supervisor）



**Part 2 服务器端接口API**

|  |
| --- |
| 共26个接口API  **2.1.admins** |
| 1. admins/getAdminsByCode  。 参数：Admins对象  。 返回值：Admins对象  。 功能：根据管理员编码和密码进行查询，用于登录  **2.2.aqi** |
| 1. aqi/listAqiAll 。 参数：无  。 返回值： List**<**Aqi**>**  。 功能：全查询AQI信息  **2.3.aqiFeedback** |

1. aqiFeedback/saveAqiFeedback

。 参数：AqiFeedback对象 。 返回值： int

。 功能：保存公众监督员的AQI反馈信息 2. aqiFeedback/listAqiFeedbackByTelId

。 参数：AqiFeedback对象

。 返回值： List<AqiFeedback>

。 功能：保存公众监督员的AQI反馈信息

3. aqiFeedback/listAqiFeedbackPage 。 参数：AfPageRequestDto对象

。 返回值： PageResponseDto<AqiFeedback>

。 功能：查询AQI反馈信息，并有模糊查询和分页功能

4. aqiFeedback/getAqiFeedbackById

。 参数：AqiFeedback对象

。 返回值：AqiFeedback对象

。 功能：根据主键查询AQI反馈信息

5. aqiFeedback/updateAqiFeedbackAssign

。 参数：AqiFeedback对象 。 返回值： int

。 功能：根据主键更新AQI反馈信息（更新指派信息）

6. aqiFeedback/listAqiFeedbackByGmId

。 参数：AqiFeedback对象

。 返回值： List<AqiFeedback>

。 功能：根据网格员编号查询指派给网格员的AQI反馈信息

7. aqiFeedback/updateAqiFeedbackState

。 参数：AqiFeedback对象 。 返回值： int

。 功能：根据主键更新AQI反馈信息（更新状态）

**2.4.gridCity**

|  |
| --- |
| 1. gridCity/listGridCityByProvinceId  。 参数： GridCity对象  。 返回值： List<GridCity>  。 功能：根据所属省查询市信息列表  **2.5.gridMember** |
| 1. gridMember/listGridMemberByProvinceId  。 参数： GridMember对象  。 返回值： List<GridMember>  。 功能：根据省和市查询可工作状态的所属网格员信息列表  2. gridMember/getGridMemberByCodeByPass  。 参数： GridMember对象  。 返回值： GridMember对象  。 功能：根据网格员登录编码和密码进行查询，用于登录  **2.6.gridProvince** |
| 1. gridMember/getGridMemberByCodeByPass  。 参数： GridMember对象  。 返回值： GridMember对象  。 功能：根据网格员登录编码和密码登录  **2.7.statistics** |

1. statistics/saveStatistics

。 参数： Statistics对象 。 返回值： int

。 功能：保存网格员提交的AQI确认信息

2. statistics/listStatisticsPage

。 参数： StatisPageRequestDto对象

。 返回值： PageResponseDto<Statistics>

。 功能：查询确认AQI信息列表，并有模糊查询和分页功能

3. statistics/getStatisticsById 。 参数： Statistics对象

。 返回值： Statistics对象

。 功能：根据主键查询确认AQI信息

4. statistics/listProvinceItemTotalStatis

。 参数：无

。 返回值： List<ProvinceItemTotalStatisDto> 。 功能：查询省分组AQI超标累计信息

5. statistics/listAqiDistributeTotalStatis 。 参数：无

。 返回值： List<AqiDistributeTotalStatisDto>

。 功能：查询以AQI等级为分组的AQI超标分布统计信息 6. statistics/listAqiTrendTotalStatis

。 参数：无

。 返回值： List<AqiTrendTotalStatisDto>

。 功能：查询当前12个月的AQI超标趋势统计信息

7. statistics/getAqiCount

。 参数：无

。 返回值： long

。 功能：查询确认AQI数据数量

8. statistics/getAqiGoodCount

。 参数：无

。 返回值： long

。 功能：查询确认AQI良好数据数量

9. statistics/getProvinceCoverage

。 参数：无

。 返回值： String

。 功能：查询省网格的全国覆盖率

10. statistics/getCityCoverage

。 参数：无

。 返回值： String

。 功能：查询大城市网格的全国覆盖率

**2.8.supervisor**



1. supervisor/getSupervisorById 。 参数： Supervisor对象

。 返回值： Supervisor对象

。 功能：根据主键查询公众监督员，用于注册时手机号码是否存在验证

2. supervisor/saveSupervisor

。 参数： Supervisor对象 。 返回值： int

。 功能：保存公众监督员信息，用于注册

3. supervisor/getSupervisorByIdByPass

。 参数： Supervisor对象

。 返回值： Supervisor对象

。 功能：根据公众监督员手机号码和密码查询，用于登录。

**Part 3 NEPS公众监督员端APP**

|  |
| --- |
| **2.1.服务器端项目** |
| 同《Part 5 NEPM系统管理端APP》-《 2.1.服务器端项目》  **2.2.前端项目** |

**2.2.1.前端项目搭建**

[**2.2.1.1**](2.2.1.1)**.开发工具检查**

1. 检查vscode是否安装成功。

2. 检查npm安装环境： 命令行下输入： npm -v

3. 检查VueCli安装环境：命令行下输入： vue -V （注意：本工程使用VueCli5.0.1版本）

附录：

1. 安装npm：直接安装node.js （输入 "npm -v " 测试是否安装成功； 输入 node –v查看node版本）

2. 全局安装vuecli： npm install -g @vue/cli （或： npm install -g @vue/cli@5.0.1）

3. 查看当前安装的vue-cli版本：vue --version 或 vue –V

4. 卸载旧版本的vue-cli： npm uninstall vue-cli -g

5. 查看远程仓库中的版本号： npm view @vue/cli versions --json

[**2.2.1.2**](2.2.1.2)**.搭建VueCli工程**

1. 命令行下进入工作空间目录中，输入： vue create tijianfront （工程名必须小写）

2. 选择预设模板：这里选择“Manually select features”（手动选择特征）

3. 模块选取： Babel、 Router

4. 选择是否使用history 形式的路由：选择： N

5. 将依赖文件放在package.json中：选择： “in package.json”

6. 是否将当前选择保存以备下次使用：选择： N

7. 进入创建好的工程目录： cdtijianfront

8. 启动工程： npm run serve

9. 在浏览器中测试： [http://localhost:8080](http://localhost:8080/)

[**2.2.1.3**](2.2.1.3)**.添加其它依赖及配置文件**

1. 添加font-awesome与axios依赖： cnpm install font-awesome@4.7.0 --save cnpm install axios@1.1.3 --save

2. 添加图片到src的assets中。

3. 在src目录下添加common.js文件

//获取当前日期（XXXX-XX-XX）

export function getCurDate() {

let now = new Date();

let year = now.getFullYear(); let month = now.getMonth() + 1; let day = now.getDate();

month = month < 10 ? "0" + month : month; day = day < 10 ? "0" + day : day;

return year + "-" + month + "-" + day; }

//获取当前时间（XX:XX:XX）

export function getCurTime() {

let now = new Date();

let hours = now.getHours();

let minutes = now.getMinutes(); let seconds = now.getSeconds();

hours = hours < 10 ? "0" + hours : hours;

minutes = minutes < 10 ? "0" + minutes : minutes; seconds = seconds < 10 ? "0" + seconds : seconds;

return hours +":" + minutes + ":" + seconds; }

//向sessionStorage中存储一个JSON对象

export function setSessionStorage(keyStr, value) {

sessionStorage.setItem(keyStr, JSON.stringify(value));

}

//从sessionStorage中获取一个JSON对象（取不到时返回null）

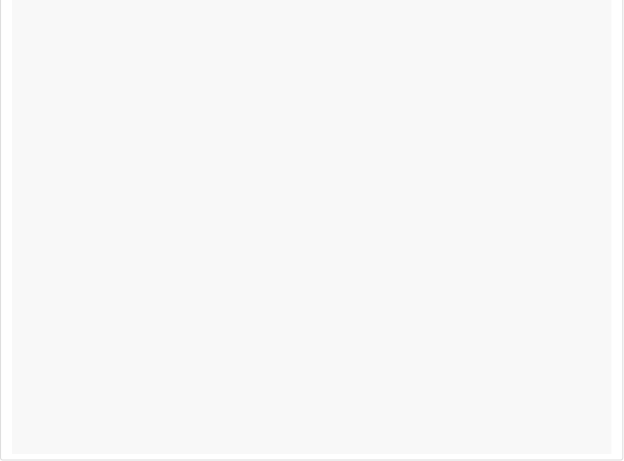
export function getSessionStorage(keyStr) { var str = sessionStorage.getItem(keyStr);

if (str == ' ' || str == null || str == 'null ' || str == undefined) { return null;

} else {

return JSON.parse(str);

} }



4. 修改vue.conﬁg.js文件，修改启动端口

const { defineConfig } = require('@vue/cli-service ')

module.exports = defineConfig({ transpileDependencies: true, devServer: {

port: 8081

} })

[**2.2.1.4**](2.2.1.4)**.main.js文件**

import { createApp } from 'vue ' import App from './App.vue '

import router from './router '

import 'font-awesome/css/font-awesome.min.css '

//vue路由守卫实现权限认证

router.beforeEach(function(to,from,next){

//从前端session中获取登录信息

let supervisor = sessionStorage.getItem('supervisor ');

//路由到“网格分配”、“预估AQI”、“历史反馈列表” ，是需要判断是否登录

if(to.path=='/selectGrid ' ||to.path=='/selectAqi ' ||to.path=='/feedback List '){ if(supervisor==null){

router.push('/login '); }

}

next(); });

createApp(App).use(router).mount('#app ')

[**2.2.1.5**](2.2.1.5)**.App.vue文件**

注意：在App.vue文件中， #app的高度也要设置为100%。

<template>

<router-view/> </template>

<script setup>

import axios from 'axios ';

import { provide } from 'vue ';

//设置共通的基础URL

axios.defaults.baseURL = 'http://localhost:8080/nepm/ '; provide('axios ',axios);

</script>

<style>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSS 重置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

html,body,div,span,h1,h2,h3,h4,h5,h6,ul,ol,li,p{ margin: 0;

padding: 0; }

html,body,#app{ width: 100%; height: 100%;

font-family: "微软雅黑 "; }

ul,ol{

list-style: none;

} a{

text-decoration: none; }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 共通总容器 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.wrapper{

width: 100%; height: 100%;

background-color: #CFE8E9; }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 共通 头部 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.header-box{

width: 100%;

}

.header-box header{ width: 100%;

height: 15.7vw;

background-color: #FFF; color: #555;

position: fixed;

left: 0; top: 0;

display: flex;

align-items: center;

justify-content: space-between;

box-sizing: border-box; padding: 0 3.6vw;

}

.header-box .top-ban{ width: 100%;

height: 15.7vw;

}

.header-box img{

display: block; width: 100%;

}

.header-box i{

font-size: 8vw;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 共通 表格表单 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.table-form{

width: 86vw;

margin: 0 auto; margin-top: 5vw;

border-collapse: collapse;

font-size: 4.2vw; }

.table-form tr td{ height: 12vw;

}

.table-form input,.table-form select,.table-form textarea{ background-color: #CFE8E9;

border: none; outline: none;

}

.table-form tr td:first-child{ color: #777;

font-weight: 700; }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 共通 按钮 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.btn{

width: 100%; height: 12vw;

background-color: #70B0BC; margin-top: 5vw;

border-radius: 2vw; color: #FFF;

font-size: 5vw;

text-align: center; line-height: 12vw; letter-spacing: 1vw; font-weight: 500;

user-select: none;

}

</style>

[**2.2.1.6**](2.2.1.6)**.路由index.js文件**

import { createRouter, createWebHashHistory } from 'vue-router ' import Login from '../views/Login.vue '

import Register from '../views/Register.vue '

import SelectGrid from '../views/SelectGrid.vue ' import SelectAqi from '../views/SelectAqi.vue '

import Feedback List from '../views/Feedback List.vue '

const routes = [ {

path: '/ ',

name: 'home ',

component: Login

},{

path: '/login ', name: 'Login ', component: Login

},{

path: '/register ', name: 'Register ', component: Register

},{

path: '/selectGrid ', name: 'SelectGrid ', component: SelectGrid

},{

path: '/selectAqi ', name: 'SelectAqi ', component: SelectAqi

},{

path: '/feedback List ', name: 'Feedback List ', component: Feedback List

} ]

const router = createRouter({

history: createWebHashHistory(), routes

})

export default router

**2.2.2.前端代码**

参考项目源代码。

**Part 4 NEPG网格员端APP**

|  |
| --- |
| **2.1.服务器端项目** |
| 同《Part 5 NEPM系统管理端APP》-《 2.1.服务器端项目》  **2.2.前端项目** |

**2.2.1.前端项目搭建**

[**2.2.1.1**](2.2.1.1)**.开发工具检查**

1. 检查vscode是否安装成功。

2. 检查npm安装环境： 命令行下输入： npm -v

3. 检查VueCli安装环境：命令行下输入： vue -V （注意：本工程使用VueCli5.0.1版本）

附录：

1. 安装npm：直接安装node.js （输入 "npm -v " 测试是否安装成功； 输入 node –v查看node版本）

2. 全局安装vuecli： npm install -g @vue/cli （或： npm install -g @vue/cli@5.0.1）

3. 查看当前安装的vue-cli版本：vue --version 或 vue –V

4. 卸载旧版本的vue-cli： npm uninstall vue-cli -g

5. 查看远程仓库中的版本号： npm view @vue/cli versions --json

[**2.2.1.2**](2.2.1.2)**.搭建VueCli工程**

1. 命令行下进入工作空间目录中，输入： vue create tijianfront （工程名必须小写）

2. 选择预设模板：这里选择“Manually select features”（手动选择特征）

3. 模块选取： Babel、 Router

4. 选择是否使用history 形式的路由：选择： N

5. 将依赖文件放在package.json中：选择： “in package.json”

6. 是否将当前选择保存以备下次使用：选择： N

7. 进入创建好的工程目录： cdtijianfront

8. 启动工程： npm run serve

9. 在浏览器中测试： [http://localhost:8080](http://localhost:8080/)

[**2.2.1.3**](2.2.1.3)**.添加其它依赖及配置文件**

1. 添加font-awesome与axios依赖： cnpm install font-awesome@4.7.0 --save cnpm install axios@1.1.3 --save

2. 添加图片到src的assets中。

3. 在src目录下添加common.js文件

//获取当前日期（XXXX-XX-XX）

export function getCurDate() {

let now = new Date();

let year = now.getFullYear(); let month = now.getMonth() + 1; let day = now.getDate();

month = month < 10 ? "0" + month : month; day = day < 10 ? "0" + day : day;

return year + "-" + month + "-" + day; }

//获取当前时间（XX:XX:XX）

export function getCurTime() {

let now = new Date();

let hours = now.getHours();

let minutes = now.getMinutes(); let seconds = now.getSeconds();

hours = hours < 10 ? "0" + hours : hours;

minutes = minutes < 10 ? "0" + minutes : minutes; seconds = seconds < 10 ? "0" + seconds : seconds;

return hours +":" + minutes + ":" + seconds; }

//向sessionStorage中存储一个JSON对象

export function setSessionStorage(keyStr, value) {

sessionStorage.setItem(keyStr, JSON.stringify(value));

}

//从sessionStorage中获取一个JSON对象（取不到时返回null）

export function getSessionStorage(keyStr) { var str = sessionStorage.getItem(keyStr);

if (str == ' ' || str == null || str == 'null ' || str == undefined) { return null;

} else {

return JSON.parse(str);

} }

//从sessionStorage中移除一个JSON对象

export function removeSessionStorage(keyStr) { sessionStorage.removeItem(keyStr);

}

//向localStorage中存储一个JSON对象

export function setLocalStorage(keyStr, value) {

localStorage.setItem(keyStr, JSON.stringify(value));

}

//从localStorage中获取一个JSON对象（取不到时返回null）

export function getLocalStorage(keyStr) { var str = localStorage.getItem(keyStr);

if (str == ' ' || str == null || str == 'null ' || str == undefined) { return null;

} else {

return JSON.parse(str); }

}

//从localStorage中移除一个JSON对象

export function removeLocalStorage(keyStr) { localStorage.removeItem(keyStr);

}

4. 修改vue.conﬁg.js文件，修改启动端口

const { defineConfig } = require('@vue/cli-service ')

module.exports = defineConfig({ transpileDependencies: true, devServer: {

port: 8083

} })

[**2.2.1.4**](2.2.1.4)**.main.js文件**

import { createApp } from 'vue ' import App from './App.vue '

import router from './router '

import 'font-awesome/css/font-awesome.min.css '

//vue路由守卫实现权限认证

router.beforeEach(function(to,from,next){

//从前端session中获取登录信息

let gridMember = sessionStorage.getItem('gridMember ');

//路由到“网格分配”、“预估AQI”、“历史反馈列表” ，是需要判断是否登录

if(to.path=='/feedback List ' ||to.path=='/feedbackInfo '){

if(gridMember==null){

router.push('/login ');

} }

next();

});

createApp(App).use(router).mount('#app ')

[**2.2.1.5**](2.2.1.5)**.App.vue文件**

注意：在App.vue文件中， #app的高度也要设置为100%。

<template>

<router-view/> </template>

<script setup>

import axios from 'axios ';

import { provide } from 'vue ';

//设置共通的基础URL

axios.defaults.baseURL = 'http://localhost:8080/nepm/ '; provide('axios ',axios);

</script>

<style>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSS 重置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

html,body,div,span,h1,h2,h3,h4,h5,h6,ul,ol,li,p{ margin: 0;

padding: 0; }

html,body,#app{ width: 100%; height: 100%;

font-family: "微软雅黑 "; }

ul,ol{

list-style: none; }

a{

text-decoration: none; }

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 头部共通样式 \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.header-box{

width: 100%;

}

.header-box header{ width: 100%;

height: 15.7vw;

background-color: #FFF; color: #555;

display: flex;

align-items: center;

justify-content: space-between;

box-sizing: border-box; padding: 0 3.6vw;

position: fixed; left: 0;

top: 0; }

.header-box .top-ban{ width: 100%;

height: 15.7vw;

}

.header-box img{

width: 100%;

display: block; height: 200px;

}

.header-box .fa{

font-size: 8vw;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 底部共通样式 \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

footer{

width: 100%;

position: absolute; left: 0;

bottom: 0; }

footer img{

width: 100%;

display: block; height: 100px;

}

</style>

[**2.2.1.6**](2.2.1.6)**.路由index.js文件**

import { createRouter, createWebHashHistory } from 'vue-router ' import Login from '../views/Login.vue '

import Feedback List from '../views/Feedback List.vue ' import FeedbackInfo from '../views/FeedbackInfo.vue '

const routes = [ {

path: '/ ',

name: 'home ',

component: Login

}, {

path: '/login ', name: 'Login ', component: Login

}, {

path: '/feedback List ', name: 'Feedback List ', component: Feedback List

}, {

path: '/feedbackInfo ', name: 'FeedbackInfo ', component: FeedbackInfo

} ]

const router = createRouter({

history: createWebHashHistory(), routes

})

export default router

**2.2.2.前端代码**

参考项目源代码。

**Part 5 NEPM系统管理端APP**



**2.1.服务器端项目**



**2.1.1.服务器端项目搭建**

[**2.1.1.1**](2.1.1.1)**.开发工具检查**

1. 开发工具： SpringToolSuite4+（ STS）

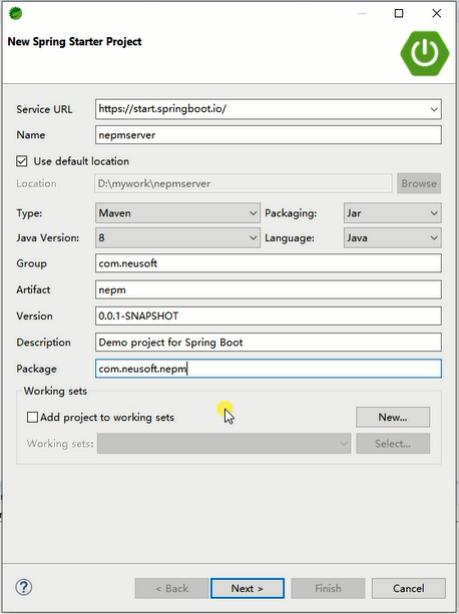
2. 检查开发工具的jdk配置：jdk8

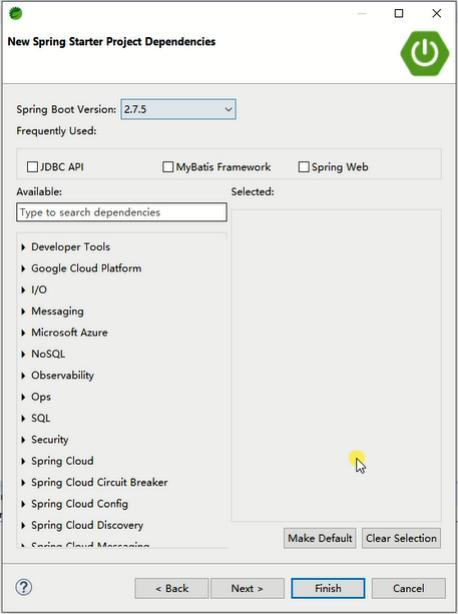
3. 检查maven构建工具的配置： maven3+

4. 检查开发工具的文件编码配置： utf-8

[**2.1.1.2**](2.1.1.2)**.搭建SpringBoot工程**

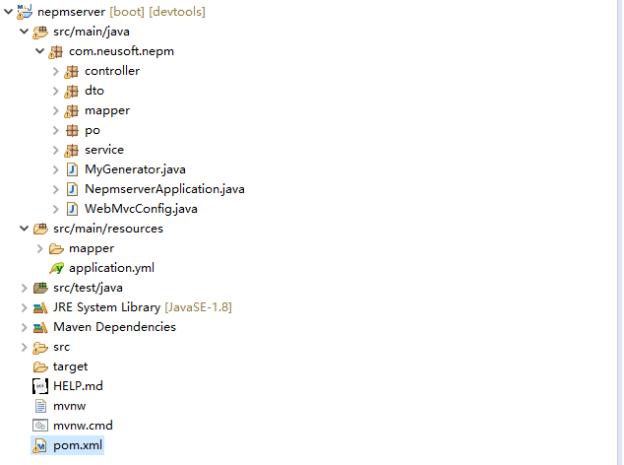
1. 工程类型：





2. pom.xml文件

3. 工程目录结构



4. SpringBoot入口文件

5. 处理跨域的Cors配置文件

6. application.yml配置文件

**2.1.2.服务器端代码**

参考项目源代码。

**2.2.前端项目**



**2.2.1.前端项目搭建**

[**2.2.1.1**](2.2.1.1)**.开发工具检查**

1. 检查vscode是否安装成功。

2. 检查npm安装环境： 命令行下输入： npm -v

3. 检查VueCli安装环境：命令行下输入： vue -V （注意：本工程使用VueCli5.0.1版本）

附录：

1. 安装npm：直接安装node.js （输入 "npm -v " 测试是否安装成功； 输入 node –v查看node版本）

2. 全局安装vuecli： npm install -g @vue/cli （或： npm install -g @vue/cli@5.0.1）

3. 查看当前安装的vue-cli版本：vue --version 或 vue –V

4. 卸载旧版本的vue-cli： npm uninstall vue-cli -g

5. 查看远程仓库中的版本号： npm view @vue/cli versions --json

[**2.2.1.2**](2.2.1.2)**.搭建VueCli工程**

1. 命令行下进入工作空间目录中，输入： vue create tijianfront （工程名必须小写）

2. 选择预设模板：这里选择“Manually select features”（手动选择特征）

3. 模块选取： Babel、 Router、Vuex

4. 选择是否使用history 形式的路由：选择： N

5. 将依赖文件放在package.json中：选择： “in package.json”

6. 是否将当前选择保存以备下次使用：选择： N

7. 进入创建好的工程目录： cdtijianfront

8. 启动工程： npm run serve

9. 在浏览器中测试： [http://localhost:8080](http://localhost:8080/)

[**2.2.1.3**](2.2.1.3)**.添加其它依赖及配置文件**

1. 添加font-awesome与axios依赖： cnpm install element-plus@2.2.17 --save cnpm install @element-plus/icons-vue@2.0.10 --save

cnpm install axios@1.1.3 --save

2. 添加图片到src的assets中。

3. 在src目录下添加common.js文件

//获取当前日期（XXXX-XX-XX）

export function getCurDate() {

let now = new Date();

let year = now.getFullYear(); let month = now.getMonth() + 1; let day = now.getDate();

month = month < 10 ? "0" + month : month; day = day < 10 ? "0" + day : day;

return year + "-" + month + "-" + day; }

//获取当前时间（XX:XX:XX）

export function getCurTime() { let now = new Date();

let hours = now.getHours();

let minutes = now.getMinutes(); let seconds = now.getSeconds();

hours = hours < 10 ? "0" + hours : hours;

minutes = minutes < 10 ? "0" + minutes : minutes; seconds = seconds < 10 ? "0" + seconds : seconds;

return hours +":" + minutes + ":" + seconds; }

//向sessionStorage中存储一个JSON对象

export function setSessionStorage(keyStr, value) {

sessionStorage.setItem(keyStr, JSON.stringify(value));

}

//从sessionStorage中获取一个JSON对象（取不到时返回null）

export function getSessionStorage(keyStr) { var str = sessionStorage.getItem(keyStr);

if (str == ' ' || str == null || str == 'null ' || str == undefined) { return null;

} else {

return JSON.parse(str);

} }

//从sessionStorage中移除一个JSON对象

export function removeSessionStorage(keyStr) { sessionStorage.removeItem(keyStr);

}

//向localStorage中存储一个JSON对象

export function setLocalStorage(keyStr, value) {

localStorage.setItem(keyStr, JSON.stringify(value));

}

//从localStorage中获取一个JSON对象（取不到时返回null）

export function getLocalStorage(keyStr) { var str = localStorage.getItem(keyStr);

if (str == ' ' || str == null || str == 'null ' || str == undefined) { return null;

} else {

return JSON.parse(str); }

}

//从localStorage中移除一个JSON对象

export function removeLocalStorage(keyStr) { localStorage.removeItem(keyStr);

}

4. 修改vue.conﬁg.js文件，修改启动端口

const { defineConfig } = require('@vue/cli-service ')

module.exports = defineConfig({ transpileDependencies: true, devServer: {

port: 8082

} })

[**2.2.1.4**](2.2.1.4)**.main.js文件**

import { createApp } from 'vue ' import App from './App.vue '

import router from './router ' import store from './store '

//导入element-plus框架

import ElementPlus from 'element-plus ';

import 'element-plus/theme-chalk/index.css ';

//使用路由守卫实现登录权限验证

router.beforeEach(function(to,from,next){

let admins = sessionStorage.getItem('admins '); if(!(to.path=='/ ' ||to.path=='/login ')){

if(admins==null){

router.push('/login ');

} }

next();

});

createApp(App).use(store).use(router).use(ElementPlus).mount('#app ')

[**2.2.1.5**](2.2.1.5)**.App.vue文件**

注意：在App.vue文件中， #app的高度也要设置为100%。

<template>

<router-view />

</template>

<script setup>

import axios from "axios";

import { provide } from "vue";

import { ElMessage } from "element-plus";

//设置共通axios基础URL

axios.defaults.baseURL = "http://localhost:8080/nepm/";

//封装共通消息提示框

const myElMessage = (message,type) => { ElMessage({

message: message, type: type,

showClose: true });

};

provide("axios", axios);

provide("myElMessage", myElMessage);

</script>

<style>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSS重置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

html, body, div,

span, h1, h2, h3, h4, h5, h6, ul, ol, li, p {

margin: 0; padding: 0;

}

html, body, #app {

width: 100%; height: 100%;

font-family: "微软雅黑 ";

}

ul, ol {

list-style: none; }

a {

text-decoration: none; }

</style>

[**2.2.1.6**](2.2.1.6)**.路由index.js文件**

import { createRouter, createWebHashHistory } from 'vue-router ' import Login from '../views/Login.vue '

import Index from '../views/Index.vue '

import HelloWorld from '../components/HelloWorld.vue '

import Feedback List from '../views/feedback/Feedback List.vue '

import StatisticsList from '../views/feedback/StatisticsList.vue ' import StatisticsInfo from '../views/feedback/StatisticsInfo.vue ' import FeedbackInfo from '../views/feedback/FeedbackInfo.vue '

import AssignFeedback from '../views/feedback/AssignFeedback.vue '

import ProvinceItemTotalStatis from '../views/statistics/ProvinceItemTotalStatis.vue ' import AqiDistributeTotalStatis from '../views/statistics/AqiDistributeTotalStatis.vue ' import AqiTrendTotalStatis from '../views/statistics/AqiTrendTotalStatis.vue '

import OtherTotalStatis from '../views/statistics/OtherTotalStatis.vue '

const routes = [ {

path: '/ ',

name: 'home ',

component: Login

},{

path: '/login ', name: 'Login ', component: Login

},{

path: '/index ', name: 'Index ',

component: Index, children: [

{

path: '/index ',

component: HelloWorld },{

path: '/index/feedback List ',

component: Feedback List },{

path: '/index/statisticsList ',

component: StatisticsList },{

path: '/index/statisticsInfo ',

component: StatisticsInfo },{

path: '/index/provinceItemTotalStatis ',

component: ProvinceItemTotalStatis },{

path: '/index/aqiDistributeTotalStatis ',

component: AqiDistributeTotalStatis },{

path: '/index/aqiTrendTotalStatis ',

component: AqiTrendTotalStatis },{

path: '/index/otherTotalStatis ',

component: OtherTotalStatis },{

path: '/index/feedbackInfo ',

component: FeedbackInfo },{

path: '/index/assignFeedback ',

component: AssignFeedback }

] }

]

const router = createRouter({

history: createWebHashHistory(), routes

})

export default router

[**2.2.1.7**](2.2.1.7)**.Vuex的index.js文件**

import { createStore } from 'vuex '

export default createStore({ state: {

maxPageNum: 10 },

getters: { },

mutations: { },

actions: { },

modules: {

} })

**2.2.2.前端代码**

参考项目源代码。

**Part 6 NEPV决策者端APP**

|  |
| --- |
| **2.1.服务器端项目** |
| 同《Part 5 NEPM系统管理端APP》-《 2.1.服务器端项目》  **2.2.前端项目** |

**2.2.1.前端项目搭建**

[**2.2.1.1**](2.2.1.1)**.开发工具检查**

1. 检查vscode是否安装成功。

2. 检查npm安装环境： 命令行下输入： npm -v

3. 检查VueCli安装环境：命令行下输入： vue -V （注意：本工程使用VueCli5.0.1版本）

附录：

1. 安装npm：直接安装node.js （输入 "npm -v " 测试是否安装成功； 输入 node –v查看node版本）

2. 全局安装vuecli： npm install -g @vue/cli （或： npm install -g @vue/cli@5.0.1）

3. 查看当前安装的vue-cli版本：vue --version 或 vue –V

4. 卸载旧版本的vue-cli： npm uninstall vue-cli -g

5. 查看远程仓库中的版本号： npm view @vue/cli versions --json

[**2.2.1.2**](2.2.1.2)**.搭建VueCli工程**

1. 命令行下进入工作空间目录中，输入： vue create tijianfront （工程名必须小写）

2. 选择预设模板：这里选择“Manually select features”（手动选择特征）

3. 模块选取： Babel、 Router

4. 选择是否使用history 形式的路由：选择： N

5. 将依赖文件放在package.json中：选择： “in package.json”

6. 是否将当前选择保存以备下次使用：选择： N

7. 进入创建好的工程目录： cdtijianfront

8. 启动工程： npm run serve

9. 在浏览器中测试： [http://localhost:8080](http://localhost:8080/)

[**2.2.1.3**](2.2.1.3)**.添加其它依赖及配置文件**

1. 添加font-awesome与axios依赖： cnpm install font-awesome@4.7.0 --save

cnpm install axios@1.1.2 --save cnpm install echarts@5.4.0 --save cnpm install jquery@1.11.3 --save

2. 添加图片到src的assets中。

3. 添加china.js文件到src的assets中。

4. 修改vue.conﬁg.js文件，修改启动端口

const { defineConfig } = require('@vue/cli-service ')

module.exports = defineConfig({ transpileDependencies: true, devServer: {

port: 8084

} })

[**2.2.1.4**](2.2.1.4)**.main.js文件**

import { createApp } from 'vue ' import App from './App.vue '

import router from './router '

import 'font-awesome/css/font-awesome.min.css '; createApp(App).use(router).mount('#app ')

[**2.2.1.5**](2.2.1.5)**.App.vue文件**

注意：在App.vue文件中， #app的高度也要设置为100%。

<template>

<router-view/> </template>

<style>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CSS 重置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

html,body,div,span,h1,h2,h3,h4,h5,h6,ul,ol,li,p{ margin: 0;

padding: 0; }

html,body,#app{ width: 100%; height: 100%;

font-family: "微软雅黑 "; }

ul,ol{

list-style: none; }

a{

text-decoration: none; }

</style>

[**2.2.1.6**](2.2.1.6)**.路由index.js文件**

import { createRouter, createWebHashHistory } from 'vue-router ' import Index from '../views/Index.vue '

const routes = [ {

path: '/ ',

name: 'home ',

component: Index },{

path: '/index ', name: 'Index ',

component: Index }

]

const router = createRouter({

history: createWebHashHistory(), routes

})

export default router

**2.2.2.前端代码**

参考项目源代码。