# March\_Half

2020-09-25

下载mysql、navicat、idea

Mysql：net start mysql（stop）

使用idea创建maven项目March\_Half

Spring：

<https://spring.io/guides/gs/spring-boot/>

<https://www.docs4dev.com/docs/zh/spring-boot/1.5.9.RELEASE/reference/getting-started-first-application.html#getting-started-first-application-code>

引入包

Mybatis：

<https://mybatis.org/mybatis-3/zh/index.html>

2020-09-26

找寻b站视频

创建march\_half\_wx

2020-09-27

调整微信小程序绑定页面

2020-9-28

调整为小程序绑定页面css

下载vue node npm

项目移入github

2020-09-29

创建march\_half\_vue

构建vue项目

2020-09-30

构建vue3.0项目（https://www.cnblogs.com/yangsg/p/13732415.html）

Vue3.0 ：

// 在vue2中 data 在vue3中使用 reactive代替

import { reactive, computed } from 'vue'

import { useRouter } from 'vue-router'

// setup相当于vue2.0的 beforeCreate和 created，是vue3新增的一个属性，所有的操作都在此属性中完成

// 通过reactive 可以初始化一个可响应的数据，与Vue2.0中的Vue.observer很相似

// 使用计算属性生成

// 在Vue3.0中，所有的数据和方法都通过在setup 中 return 出去

//provide inject 父子

* ~~beforeCreate~~ -> use setup()
* ~~created~~ -> use setup()
* beforeMount -> onBeforeMount
* mounted -> onMounted
* beforeUpdate -> onBeforeUpdate
* updated -> onUpdated
* beforeUnmount -> onBeforeUnmount
* unmounted -> onUnmounted
* errorCaptured -> onErrorCaptured
* renderTracked -> onRenderTracked
* renderTriggered -> onRenderTriggered

小小的todolist vue3 demo

继续mybatis

<https://blog.csdn.net/iku5200/article/details/82856621>



创建interface跟对应的映射xml：

右击类名copy referen --> xml文件namespace

方法名getUserByLoginName copy 🡪 xml文件id

初始化 开始连接微信小程序

成功实现登录

2020-10-01

重新创建idea项目

希望能微信小程序绑定系统账号

<https://my.oschina.net/u/3337958/blog/1618224>

<https://my.oschina.net/u/3337958/blog/1618405>

授权登录 获取每个微信的独特openid

2020-10-03

创建vue2.0项目制作后端  
 尝试在2.0导入npm install @vue/composition-api --save

2020-10-04

绑定信息页面 想实现增删改查

个人页面的信息修改

整体的修改了关于用户的数据表

根据这个实现了增加跟全体遍历

2020-10-05

实现登陆账号根据openid增删改查

开始准备实现home首页的内容

并开始后台页面

Api（axios）、拦截器（request.js）---封装axios

<http://www.axios-js.com/zh-cn/docs/#%E6%8B%A6%E6%88%AA%E5%99%A8>

2020-10-06

Vue3实现axios跨域

但是没有服务器还是尝试自己本地的数据库

Vue与spring之间的跨域一直不成功

2020-10-07

经过检查发现是vue的数值传递出现了问题，在get的情况下不应该使用date而是params。

了解了一下axios的get跟post 发现可以实现get在url里传递数值，还没有实现post的通过request body传递参数。

截至现在，基本实现微信小程序、vue、spring之间的增删改查，还有复杂的数据操作还没有开始。

接下来就是开始设计程序要实现哪些功能，做一个整体的规划。

希望能在时间跟精力至于可以实现更多方面。

2020-10-08

2020-10-09

改变了自己vue后台的背景图片

思考自己这个小程序大概需要实现那几个功能，开始设计自己的个人页面还有首页，初步考虑想用多彩的卡片式

（微信小程序实现卡片层叠滑动效果）

<https://www.jb51.net/article/163592.htm>

（css实现梯形（各种形状）的网页布局——transform的妙用）

<https://www.cnblogs.com/pengfei-nie/p/6919687.html>

删去最开始给个人页面设计的半圆形，改成梯形

改变自己的# 颜色配色，打算使用白色为主色调，采用多彩的卡片

2020-10-10

开始翻看论文，寻找文献参考

2020-10-11

敲完任务书的概况

实现个人页面的改版和绑定信息的页面，还有新建首页

使用echart制作图表

Rgb-16转换<https://www.sioe.cn/yingyong/yanse-rgb-16/>

在线取色器<http://www.jiniannet.com/Page/allcolor>

2020-10-12

继续后台的首页搭建





开启后台页面的首页，敲定颜色

构思结构，也大概确定下数据库的建表。

2020-10-14

重建后台首页，并创建好相对应的导航栏

研究了sessionstorage和localStorage：

localstorage是本地存贮 ，清除本地存贮才可以。sessionstorage 是关闭浏览器/页面 数据就会清除。

（cookie 一般设定是20min 但是你也可以设置 记住： 它只在设定时间前有效）

localStorage是一种比cookie大的存贮空间。 大约5m

四种方法 ：1 localStorage.setItem(); 传

2. localStorage.getItem(); 接

3 localStorage.removeItem() ; 删

4 localStorage.setItem(); 改

sessionStorage

sessionStorage.key(int index) //返回当前 sessionStorage 对象的第index序号的key名称。若没有返回null。

sessionStorage.getItem(string key) //返回键名(key)对应的值(value)。若没有返回null。

sessionStorage.setItem(string key, string value) //该方法接受一个键名(key)和值(value)作为参数，将键值对添加到存储中；如果键名存在，则更新其对应的值。

sessionStorage.removeItem(string key) //将指定的键名(key)从 sessionStorage 对象中移除。

sessionStorage.clear() //清除 sessionStorage 对象所有的项。

Json：// 存储值：将对象转换为Json字符串

sessionStorage.setItem('user', JSON.stringify(userEntity));

// 取值时：把获取到的Json字符串转换回对象

var userJsonStr = sessionStorage.getItem('user');

userEntity = JSON.parse(userJsonStr);

2020-10-15

微调整后台页面的main框架

创建数据库types表，并在spring中创建好相对应的xml、dao、type、typeControll文件，实现类别的增删改查。而在后台页面中，加载出来所有类别。

（在此之前，基本完成两个平台的登录注册问题）

2020-10-16

傍晚解决昨天遗留下来的打开dialog会先灰屏的问题，:append-to-body="true"

2020-10-17

2020-10-18

开始在后台页面加上type的增删改查功能，并同样的样式完成了管理员的增删改查

2020-10-19

回忆之前做的nodejs项目

2020-10-20

边回忆边继续开始后台页面的图片上传功能

2020-10-21

执行任务

2020-10-22

继续努力实现图片上传

Tinymce组件  
 <https://www.cnblogs.com/qinshengfei/p/12307423.html>

<https://blog.csdn.net/weixin_40337982/article/details/84031778>

<https://www.cnblogs.com/aknife/p/11840347.html>

<https://blog.csdn.net/qq_42852004/article/details/90055999>

<https://www.cnblogs.com/wisewrong/p/8985471.html>

<https://blog.csdn.net/weixin_43809369/article/details/90212044>

2020-10-23

<https://liubing.me/vue-tinymce-5.html>

根据教程展示出了tinymce富文本，使用2的语法写了一个简单组件，可以随时调用。

图片上传是个极大的难点，特别是如何前端获取图片之后，如何传给后端。后端又如何将图片返回到前端预览。

<https://www.52pojie.cn/thread-822874-1-1.html>

<https://blog.csdn.net/qq_41618510/article/details/95199806>

<https://blog.csdn.net/zhizungege/article/details/106839214>

2020-10

十月底完成了任务书

2020-11

十一月初完成了外文翻译

2020-11-09

开始开题报告

<https://wenku.baidu.com/view/9decc5b16aec0975f46527d3240c844769eaa082.html>

<https://www.baywatch.cn/wenhualunwen/qinshiwenhualunwen/150880.html>

<https://wenku.baidu.com/view/9606d7415727a5e9856a61cc.html>

<https://wenku.baidu.com/view/8ba1561c964bcf84b9d57bf4.html>

<https://www.ixueshu.com/document/fc3df61d43eb5809318947a18e7f9386.html>

<http://www.szjianhe.com/2014/0630/87.html>

<https://www.sohu.com/a/335969740_120076378>

<https://max.book118.com/html/2018/0820/6103044221001212.shtm>

看到一个好看的后台

[www.euiadmin.com](http://www.euiadmin.com)

2020-11-10

<https://blog.csdn.net/qq_41786117/article/details/90046144>

<https://wenku.baidu.com/view/5f8e572331b765ce050814fc.html>

<https://wenku.baidu.com/view/c415f092daef5ef7ba0d3ce2.html>

2020-11-13

<https://wenku.baidu.com/view/77fc37150a1c59eef8c75fbfc77da26924c59619.html>

2020-11-15

<https://wenku.baidu.com/view/53ed69b6b80d6c85ec3a87c24028915f814d84d5.html>

<http://www.lwfree.com/yanjiu/lunwen_33699.html>

<http://www.lwfree.com/yanjiu/lunwen_9539.html>

<https://xueshu.baidu.com/usercenter/paper/show?paperid=2a8df5cf33672ce6f6e94f9470078896&site=xueshu_se>

<https://xueshu.baidu.com/usercenter/paper/show?paperid=36f0f0dd6681f289334dab092edfbf23&site=xueshu_se>

<https://xueshu.baidu.com/usercenter/paper/show?paperid=5e90377bd3715e79157fdddf8aadbf1a&site=xueshu_se>

<https://wenku.baidu.com/view/c1e6e67cec630b1c59eef8c75fbfc77da269970e.html>

2020-11-17

<https://xueshu.baidu.com/usercenter/paper/show?paperid=964eed97b55cbfad9f066ab5d06f3e9e&site=xueshu_se>

<https://xueshu.baidu.com/usercenter/paper/show?paperid=2a8df5cf33672ce6f6e94f9470078896&site=xueshu_se>

<https://xueshu.baidu.com/usercenter/paper/show?paperid=36f0f0dd6681f289334dab092edfbf23&site=xueshu_se>

2020-11-18

解决之前spring后端接收不到vue前端用post传递值的问题，在vue前端的api中加入var qs = require('querystring')，qs.stringify(data)

2020-11-19

<template>

    <div id="home">

        <el-container style="height:100vh;" v-loading="loading">

            <el-header style="text-align: right; font-size: 12px">

                <el-col :span="20">

                   <div class="march">

                       MARCH\_HALF

                   </div>

                </el-col>

                <el-col :span="4">

                   <el-dropdown>

                    <i class="el-icon-setting" style="margin-right: 15px"></i>

                    <el-dropdown-menu slot="dropdown">

                        <el-dropdown-item>修改密码</el-dropdown-item>

                        <el-dropdown-item>个人信息</el-dropdown-item>

                        <el-dropdown-item>退出系统</el-dropdown-item>

                    </el-dropdown-menu>

                </el-dropdown>

                <span>{{currentPerson.nickname}}</span>

                </el-col>

            </el-header>

            <el-container>

                <el-aside width="200px" style="background-color: rgb(238, 241, 246)">

                    <el-menu router default-active="/borad"  :default-openeds="['1','2','3']">

                        <el-menu-item index="/borad"><i class="el-icon-s-marketing"></i>看板</el-menu-item>

                        <el-submenu index="2">

                            <template slot="title"><i class="el-icon-menu"></i>基础数据</template>

                            <el-menu-item-group>

                                <template slot="title">基本</template>

                                <el-menu-item index="/dorm">寝室管理</el-menu-item>

                                <el-menu-item index="/user">人员管理</el-menu-item>

                            </el-menu-item-group>

                            <el-menu-item-group title="论坛">

                                <el-menu-item index="/post">帖子管理</el-menu-item>

                            </el-menu-item-group>

                            <el-submenu index="1">

                                <template slot="title">首页</template>

                                <el-menu-item index="/notice">公告管理</el-menu-item>

                                <el-menu-item index="/pay">缴费管理</el-menu-item>

                                <el-menu-item index="/hygiene">卫生管理</el-menu-item>

                                <el-menu-item index="/repair">报修管理</el-menu-item>

                                <el-menu-item index="/out">外出管理</el-menu-item>

                                <el-menu-item index="/act">活动管理</el-menu-item>

                            </el-submenu>

                        </el-submenu>

                        <el-submenu index="3">

                            <template slot="title"><i class="el-icon-s-tools"></i>基础设置</template>

                            <el-menu-item index="/type">类别</el-menu-item>

                            <el-menu-item index="/power">权限</el-menu-item>

                            <el-menu-item index="/admin">管理员</el-menu-item>

                        </el-submenu>

                    </el-menu>

                </el-aside>

                <el-main>

                    <router-view></router-view>

                </el-main>

            </el-container>

        </el-container>

    </div>

</template>

<script>

    import {

        onMounted,

        reactive,

        ref,

    } from "@vue/composition-api";

    import request from "@/utils/request";

    export default {

        name: 'home',

        setup(props, {

            refs,

            root

        }) {

            let loading = ref(true);

            let currentPerson = reactive({

                userName:'',

                nickname:'',

                password:''

            });

            const userData = (() => {

                // var rootStr = sessionStorage.getItem('root');

                // var rootJson = JSON.parse(rootStr);

                currentPerson.userName = sessionStorage.getItem('userName');

                currentPerson.password = sessionStorage.getItem('password');

                currentPerson.nickname = sessionStorage.getItem('nickname');

                loading.value = false;

            });

            onMounted(() => {

                userData();

            })

            return {

                loading,

                userData,

                currentPerson

            }

        }

    };

</script>

<style scoped>

    #home,

    section {

        height: 100%;

    }

    #home {

        /\* background: url("../../assets/imgs/summer.jpg"); \*/

        width: 100%;

        height: 100%;

        position: fixed;

        background-size: 100% 100%;

    }

    .march{

        font: 2rem sans-serif;

        text-align: center;

        line-height: 60px;

    }

    .el-header {

        background-color: #B3C0D1;

        color: #333;

        line-height: 60px;

    }

    .el-aside {

        color: #333;

    }

    .el-menu {

        height: 100%;

    }

    .el-main {

        padding: 0;

    }

</style>

根据B站教学，重改home首页布局，使用组件header\main\main

Scss格式文件，全局使用变量。





实现一个全局组件。

2020-11-20

Es6的模板语法不能使用‘’

完善tinymce富文本编辑器

2020-11-21

使用vuex组件通信

2020-11-22

用按钮通过vuex来控制nav导航栏的缩放



2020-11-24

使用VUEX的store来全局存储变量。一般系统变量放在app中，并将app用模块思想引入index。

使用cookie。Premit文件中利用token来路由守卫。在utiles/app文件中写入一般使用的cook控制。

在没有token的情况下，只会定向到login页面。

实现进入login页面，清除用户数据（包括点击退出按钮 ）

2020-11-25

实现退出系统按钮的功能，用promise来操作。

removeToken();

    removeNickName();

    content.commit('SET\_TOKEN',"");

    content.commit('SET\_NICKNAME',"");

在mian.js中加入这段全局，来解决nav中点击没有改变的问题。

import Router from 'vue-router'

const originalPush = Router.prototype.push

Router.prototype.push = function push(location) {

  return originalPush.call(this, location).catch(err => err)

}

调整api文件、分页功能。

<https://blog.csdn.net/weixin_45777544/article/details/102669607>

<https://blog.csdn.net/zhouzhiwengang/article/details/109972560>

<https://www.cnblogs.com/kadima-zy/p/page1.html>

<https://liuyanzhao.com/10373.html>

2020-11-27

<https://blog.csdn.net/weixin_42601136/article/details/109152431?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-10.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-10.control>

idea报错Diamond types are not supported at language level '6'

<https://blog.csdn.net/qq_33188180/article/details/79147065?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromBaidu-3.control>

service = dao

entity = pojo

*#pagehelper:  
# helper-dialect: mysql  
# reasonable: true  
# support-methods-arguments: true  
# params:*

<!-- <dependency>-->  
<!-- <groupId>com.github.pagehelper</groupId>-->  
<!-- <artifactId>pagehelper</artifactId>-->  
<!-- <version>5.1.2</version>-->  
<!-- </dependency>-->  
<!-- <dependency>-->  
<!-- <groupId>com.github.pagehelper</groupId>-->  
<!-- <artifactId>pagehelper-spring-boot-starter</artifactId>-->  
<!-- <version>1.2.3</version>-->  
<!-- </dependency>-->

<https://www.cnblogs.com/zhangliwei/p/9575867.html>

分页插件导入失败，暂时理由应该是版本不适配（理由就是版本不适配）

解决版本适配，简单的操作了一下分页

2020-11-28

axios.get不能用@RequestBody获取，能用@RequestParam获取。

axios.post 可以用@RequestBody获取且不需要qs.stringify序列化，直接传对象就可以了。但如果post请求后台用@RequestParam()获取的时候，前端需要用qs.stringify序列化所传递的参数

目前问题就还是后端RequestParam，接收不到前端传输的值

查找之后发现问题出在变量名的不一致。

静态数据的格式写法

    //table.loading = true;

    // setTimeout(function () {

    //     table.loading = false;

    //     var temp = [{

    //             projectName: '测斜',

    //             dotNumNow: 'p17(17.5m)',

    //             variationNow: '1.45(mm)',

    //             dotNumTotal: 'P11(11.5m)',

    //             variationTotal: '63.42(mm)',

    //             data: '2020-10-01'

    //         },

    //         {

    //             projectName: '地表沉降',

    //             dotNumNow: 'D08-2',

    //             variationNow: '-0.78(mm)',

    //             dotNumTotal: 'D02-2',

    //             variationTotal: '-11.86(mm)',

    //             data: '2020-10-02'

    //         }

    //     ]

    //     table.tableData = temp.filter(function (item) {

    //         if (form.projectVal != '') return item.projectName == form.projectVal;

    //         return item;

    //     }).filter(function (item) {

    //         if (form.dataVal != '') return item.data == form.dataVal;

    //         return item;

    //     });

    //     pagination.totalRecordCount = table.tableData.length;

    // }, 1000);

模糊查询语句该 select \* from types where typeName like "%"#{keyWord}"%"

2020-12-01

当富文本在某一个弹窗上使用时，工具栏会出现下拉选择时的层级比弹窗的小，所以，选项会被弹窗遮挡。而解决这个问题，需要把工具栏的下拉框的层级提高，找到skins/skins.min.css文件.把class名为tox-tinymce-aux的第一个的z-index属性变大即可。

Tinymce在dialog中第二次打开会有bug，使用v-if="noticeDialog.visible"。

<https://blog.csdn.net/zhizungege/article/details/106839214>

<https://www.cnblogs.com/qinshengfei/p/12307423.html>

<https://blog.csdn.net/hearth_b/article/details/106720424>

<https://segmentfault.com/a/1190000021745351>

<https://www.cnblogs.com/aknife/p/11840347.html>

<https://blog.csdn.net/weixin_40337982/article/details/84031778>