H5 开发项目流程升级改造

1. 配置方式-菜单模块配置

```
{"staticUrl":"{host}/static/gcjg_quality_field_h5/index.html#/SecurityIssueRegistrationCheckList?tenantId={tenantId}&id={orgId}&type={orgType}&userToken={token}&lang={lang}"}
```

gcjg_quality_field_h5： 仓库名称

SecurityIssueRegistrationCheckList：路由

2. 代码基础改动，兼容之前的模式

- **userInfo.js 全量覆盖**

```
export default {
    data() {
        return {
            userInfo: {},
            tenantConfig: {},
        };
    },
    created() {
        this.getCurrentUserInfo();
    },
    methods: {
        // 获取当前登录用户信息
        async getCurrentUserInfo() {
            this.userInfo = window.entryUserInfo || {};
            window['userInfo'] = this.userInfo || {};
            sessionStorage.setItem('userInfo', JSON.stringify(this.userInfo));
            return Promise.resolve(this.userInfo);
        },
    },
};
```

- **Main.js 部分改动,按需引入**

```
import Vue from 'vue';
import App from './App.vue';
import router from './router';
import store from './store';
import api from 'api/bimoneApi';
import { fetchNativeApi } from 'api/offLineConfig';
import 'utils/flexible.js';
import { getUrlVar } from 'utils/utils.js';
import { startupParameterConfig } from 'utils/startupParameterConfig.js';
import Component from 'vue-class-component';
import echarts from 'echarts';
import Echats from 'vue-echarts';
import moment from 'moment';
import { defaultCustomConfig } from 'src/tenantConfig/standard/defaultPageConfig';
import {
    gotoPage,
    replacePage,
    getRouteParams,
    resetNavInfo,
    routeBack,
} from '@/app/pageRoute';
Vue.prototype.$gotoPage = gotoPage;
Vue.prototype.$replacePage = replacePage;
Vue.prototype.$getRouteParams = getRouteParams;
Vue.prototype.$resetNavInfo = resetNavInfo;
Vue.prototype.$routeBack = routeBack;

import { setTheme } from 'src/assets/theme/theme';
// TODO 暂时全局引入，因为组件库不知道用到了哪个组件，怕有遗漏
// 组件库引入,可全量使用组件
import '../node_modules/vant/lib/index.css';
import mobileComponents from 'gcjg-mobile-components';
Vue.use(mobileComponents);
import Vant from 'vant';
Vue.use(Vant);
import './styles/index.less';
import { themes } from 'src/assets/theme/model.js';
window['themes'] = themes;
Vue.config.productionTip = false;
Component.registerHooks([
    'beforeRouteEnter', // 进入路由之前
    'beforeRouteLeave', // 离开路由之前
    'beforeRouteUpdate',
]);

Vue.prototype.$chart = echarts;
```

- **utils/startupParameterConfig.js 新增**

```
import { customPathMatch } from 'src/router/moduleConfig';
export const startupParameterConfig = (getUrlVar, api) => {
    // 组件内部使用
    if (window['bimOneApi'] && Object.keys(window['bimOneApi']).length) {
        window['bimOneApi'] = Object.assign(window['bimOneApi'], api);
    } else {
        window['bimOneApi'] = api || {};
    }
    window['env'] = process.env.NODE_ENV;
    window['getUrlVar'] = getUrlVar;
    // 全局跳转参数
    window['getModuleUrl'] = (moduleCode) => {
        const fullPath = fileNameMatchFullPath(moduleCode);
        return `{host}${
            process.env.VUE_APP_REAL_ENV !== 'dev'
                ? '/static/ytgj_jgshk_field_h5'
                : ''
    }/index.html#/${fullPath}?tenantId={tenantId}&id={orgId}&type={orgType}&userToken={token}&token={token}&lang={lang}`;
    };
};
/**
 * 文件名匹配路由全路径
 */
const fileNameMatchFullPath = (moduleCode) => {
    const pathObj = customPathMatch();
    let fullPath = moduleCode;
    for (const [key, fileObj] of Object.entries(pathObj)) {
        if (fileObj.fileNames?.some((fileName) => fileName === moduleCode)) {
            fullPath = `${key}_${moduleCode}`;
            break;
        }
    }
    return fullPath;
};
```

- **shims-tsx.d.ts 全量覆盖**

```
import Vue, { VNode } from 'vue';
declare global {
    namespace JSX {
        // tslint:disable no-empty-interface
        interface Element extends VNode {}
        // tslint:disable no-empty-interface
        interface ElementClass extends Vue {}
        interface IntrinsicElements {
            [elem: string]: any;
        }
    }
    interface Window {
        Glodon3rdNativeEventOnload: any;
        Glodon3rdNativeEvent: any;
        gldne: any;
    }
}
```

- **router/moduleConfig.js 全量覆盖**

```
/**
 * 默认基础路由
 * @returns basicRoutes
 * import 引入模式
 */
export const basicRouters = () => {
    // 1、默认路由
    const basicRoutes = [
        {
            path: '/',
            redirect: '/componentsTest',
        },
        {
            // h5 单页面
            path: '/phoneSinglePage',
            name: 'phoneSinglePage',
            component: () => import(`page/singlePage/PhoneSinglePage.vue`),
        },
        {
            path: '/componentsTest',
            name: 'componentsTest',
            component: () => import(`page/singlePage/PhoneSinglePageTest.vue`),
        },
        {
            path: '/SpecialOperator',
            name: 'SpecialOperator',
            component: () => import(`components/specialOperator/SpecialOperator.vue`),
        },
        {
            path: '/SpecialOperatorDetail',
            name: 'SpecialOperatorDetail',
            component: () =>
                import(`components/specialOperator/SpecialOperatorDetail.vue`),
        },
        {
            path: '/WorkMechanical',
            name: 'WorkMechanical',
            component: () => import(`components/workMechanical/WorkMechanical.vue`),
        },
        {
            path: '/WorkMechanicalDetail',
            name: 'WorkMechanicalDetail',
            component: () =>
                import(`components/workMechanical/WorkMechanicalDetail.vue`),
        },
    ];
    return basicRoutes;
};
```

- **Router.js 全量覆盖**

```
import Vue from 'vue';
import VueRouter, { RouteConfig } from 'vue-router';
import { basicRouters, customPathMatch, automaticPath } from './moduleConfig';
Vue.use(VueRouter);
/**
 * 1、默认写死路由
 */
const getBasicRoutes = () => {
    return basicRouters();
};
/**
 * 2、自定义路由，全路径拼写
 */
const getCustomRouter = () => {
    const routes: Array<RouteConfig> = [];
    const pathObj = customPathMatch();
    for (const key in pathObj) {
        const fileObj = pathObj[key];
        fileObj?.fileNames?.length &&
            fileObj?.fileNames.forEach((fileName) => {
                const route: RouteConfig = {
                    path: `/${key}_${fileName}`,
                    name: `${key}_${fileName}`,
                    component: () => import(`components/${fileObj.path}/${fileName}`),
                };
                routes.push(route);
            });
    }
    return routes;
};
/**
 * 3、自动路由,根据文件夹名找到下面 company/park 目录的所有组件加载
 */
const getAutomaticRouter = () => {
    const pageContext = require.context(
        `components/`,
        true,
        /(?:park|company)\/(.*?\.vue)/,
    );
    const folderNameArr = automaticPath();
    let pageRoutes: RouteConfig[] = [];
    folderNameArr?.length &&
        folderNameArr?.forEach((folderName) => {
            pageContext?.keys()?.map((id) => {
                if (id.includes(folderName)) {
                    const componentPath = id.replace(/\.\//g, '').replace(/\.vue$/, '');
                    const componentName = id
                        .split('/')
```

3. 开发模式

```
window.Glodon3rdNativeEvent &&
    window.Glodon3rdNativeEvent.openPage({
        url: window.getModuleUrl('DangerousOperationsDetail'),
        navInfo: { titleBar: { title: ' ' } },
        params: {
            item: { value: item },
        },
});
```