安全风险管理系统接口定义文档

目录

[一． 用户模块接口 3](#_Toc8825)

[1. 用户登录 3](#_Toc8826)

[2. 取得完整用户信息 3](#_Toc8827)

[二． 部门模块接口 4](#_Toc8828)

[1．递归取得所有部门.............................................................................................................4](#_Toc8829)

[三． 菜单模块接口 5](#_Toc8830)

[1．递归取得所有菜单.............................................................................................................5](#_Toc8831)

[四． 角色模块接口 5](#_Toc8832)

[1．取得所有角色.....................................................................................................................5](#_Toc8833)

[五． 岗位模块接口 6](#_Toc8834)

1．取得所有岗位.....................................................................................................................6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 服务地址说明表 | |  |
| 序号 | 请求类型 | 非法请求统一结果 | 请求根路径 | 说明 |
| 1 | post | {"result":"error"} | http://192.168.1.107:5000/ |  |

# 一． 用户模块接口

## **1.** 用户登录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | 用户登录接口 | |  |
| 序号 | 路径 | 请求类型 | 输入 | 输出 | 说明 |
| 1 | /auth/Login/LoginCheck | post | username password | Code = 0 成功 |  |
| Code = -1 失败 |

结果：

{

    "code": 0,

    "msg": "登录验证成功",

    "data": {

        "TokenValue": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1lIjoiYWRtaW4iLCJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYW1laWRlbnRpZmllciI6IjhENkQwQjBFLTU3MzktNDFBNC04RUEzLTQ2QTVGQzRBMTdFMSIsIk9yZ2FuaXplSWQiOiJDMTYyMkNCQi0wQjlBLTRFNUEtODNCRS0xMzY5MkVDMkJFMzUiLCJPcmdhbml6ZU5hbWUiOiLplKHmnpfpg63li5LljLvkv53lsYAiLCJSZWFsTmFtZSI6Iuezu-e7n-euoeeQhuWRmCIsIklzQWRtaW4iOiIxIiwiaXAiOiIxMjcuMC4wLjEiLCJuYmYiOjE1ODk0MTU4MTMsImV4cCI6MTU4OTQyNTg5MywiaXNzIjoieHkiLCJhdWQiOiJldmVyeW9uZSJ9.KursXGYUdZKCqSNvcG\_HebmNOrrukyPoQXbFxxnqJ74",

    "Expires": "2020-05-14 11:11:33",

    "TokenType": "Bearer"

    }

}

## **2.** 取得完整用户信息

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 取得用户完整信息 | |  |  |
| 序号 | 路径 | 请求类型 | 输入 |  | 输出 | 说明 |
| 1 | /api/User/GetUserInfoById | Get | userid |  | 结果 | 数组 |

结果：

{

    "count": 0,

    "code": 0,

    "msg": "查询成功",

    "data": {

        "GenderName": "女",

        "IdTypeName": "身份证",

        "LAY\_CHECKED": false,

        "UserId": "3a35204e-112b-4248-83f0-6b4e5db746b8",

        "UserName": "wdx",

        "Password": null,

        "SecretKey": null,

        "RealName": "wdx111",

        "NickName": null,

        "Gender": null,

        "Birthday": "2019-04-15",

        "IdType": null,

        "IdNumber": null,

        "MobilePhone": null,

        "OtherPhone": null,

        "OICQ": null,

        "WeChat": null,

        "Address": null,

        "OrganizeId": "98d4cd7c-11b1-4ee2-8d4f-1a9d55047daf",

        "OrganizeName": null,

        "DepId": null,

        "DepName": null,

        "Remark": "测试数据",

        "IsAdmin": 0,

        "IsNotice": 0,

        "SortCode": 1

    }

}

# 二． 系统基础管理模块接口

## **1**．用户管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/User/GetPagetListByUser | get | string orgid,string depid,  string condition, string keyword,int page,int limit |  | 获取用户列表  orgid depid 单独传入  condition:UserName/RealName  keyword 为查询关键字 |
| 2 | /api/User/Create | post | 参数 model |  | model对应与获取用户列表字段相匹配 为json字符串 |
| 3 | /api/User/Edit | Post | model |  | model对应与获取用户列表字段相匹配 为json字符串 |
| 4 | /api/User/Remove | post | keyValue |  |  |
| 5 | /api/User/RemoveBatch | post | keyValues |  |  |
| 6 | /api/User/EditCurrentUserPassword | post | newPassword  Password |  | 新密码  旧密码 只能修改当前用户 |
| 7 | /api/User/EditUserPassword | Post | userid  username  newPassword |  | 管理员可修改指定用户 |

2.行政区划管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Area/GetListToTree | get | 无 |  | 获取所以行政区划列表 树结构 |
| 2 | /api/Area/GetPageListByCondition | get | string parentId, int page, int limit |  | 点击左侧树 获取右侧列表显示  ParentId为左侧树节点的areaid |
| 3 | /api/Area/Create | Post | model |  | model 对应 AreaId ParentId AreaCode  Layer(int) SimpleSpelling SortCode Layer  Remake |
| 4 | /api/Area/Edit | post | model |  | areaid 主键必须传 |
| 5 | /api/ Area /Remove | post | keyValue |  | 主键 |
| 6 | /api/ Area /RemoveBatch | post | keyValues |  | 主键数组 |
| 7 | /api/Area/GetListByParentId | Get | ParentId |  | 获取下拉联动用 如机构管理添加行政区划 |

3.机构管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Organize/GetPageListByCondition | get | string condition, string keyword,int page,int limit |  | 获取机构列表 带分页  condition:OrgCode/OrgName/Manager  keyword 为查询关键字 |
| 2 | /api/Organize/Create | post | model |  | 与获取列表的字段相对应传入同时包含行政区划PAreaId PAreaName SAreaId SAreaName XAreaId XAreaName |
| 5 | /api/Organize /Remove | post | keyValue |  | 主键 |
| 6 | api/Organize /RemoveBatch | post | keyValues |  | 主键数组 |
| 7 | /api/Organize/GetListByCondition | get | string condition, string keyword |  | 获取机构列表 不带分页  condition:OrgCode/OrgName/Manager  keyword 为查询关键字 |

4.部门管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Department/GetPageListByCondition | get | string orgid  string condition, string keyword,int page,int limit |  | 获取部门列表 带分页  orgid 单独传入  condition:BH/Name/OrgId  keyword 为查询关键字 |
| 2 | /api/Department/Create | post | model |  | 与获取列表的字段相对应传入同时包含行政区划OrgId OrgName |
| 5 | /api/Department/Remove | post | keyValue |  | 主键 |
| 6 | /api/Department/RemoveBatch | post | keyValues |  | 主键数组 |
| 7 | /api/Department/GetListByCondition | Get | string condition, string keyword |  | 获取部门列表  condition:BH/Name  keyword 为查询关键字 |

5.系统功能菜单管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Module/GetListByCondition | get | 无 |  | 左侧树 菜单列表 |
| 2 | /api/Module/GetPageListByCondition | get | string parentId,int page,int limit |  | 点击左侧树节点 获取右侧菜单列表  parentId 为树节点id  初始化时为0 |
| 2 | /api/Module/Create | post | model |  | Model 对应的key有 ModuleName ModuleCode ParentId Icon Url Remark SortCode |
| 3 | /api/Module/Edit | post | model |  | Model 对应的key有 ModuleName ModuleCode ParentId Icon Url Remark SortCode 多了个ModuleId |
| 5 | /api/Module/Remove | post | keyValue |  | 主键 |
| 6 | /api/Module/RemoveBatch | post | keyValues |  | 主键数组 |

6.角色管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Role/GetPageListByCondition | get | string condition, string keyword,int page,int limit |  | 获取角色列表 带分页  condition:BH/Name  keyword 为查询关键字 |
| 2 | /api/Role/Create | post | model |  | 与获取列表的字段相对应传入 |
| 3 | /api/Role/Edite | post | model |  | 与获取列表的字段相对应传入 并加入主键RoleId |
| 4 | /api/Role/Remove | post | keyValue |  | 主键 |
| 5 | /api/Role/RemoveBatch | post | keyValues |  | 主键数组 |

角色成员分配

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/UserRole/GetUserList | get | string roleId,string depId |  | 根据角色Id获取该角色下已分配的用户userid depId为该部门下的该角色下的所有用户信息 |
| 2 | /api/UserRole/GetListByCondition | get | string roleId,int page,int limit |  | 获取所有用户信息 并把传入的有关roleid该角色下已分配的用户加入标识为  LAY\_CHECKED = true; |
| 3 | /api/User/GetPagetListByUser | post | string condition, string keyword,int page,int limit |  | 获取用户列表  condition:UserName/RealName/DepName  keyword 为查询关键字 |
| 4 | /api/UserRole/Authorized | post | model |  | model为json字符串 key为  RoleId 和 UserId（选中的） |

角色菜单授权

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/RoleModule/GetListByCondition | get | string roleId |  | 获取所有用户信息 并把传入的有关roleid该角色下已分配的用户菜单加入标识为  LAY\_CHECKED = true; |
| 3 | /api/UserRole/Authorized | post | model |  | model为json字符串 key为  RoleId 和 ModuleId（选中的） |

7.数据字典管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/DataDict/GetListByCondition | get | 无 |  | 左侧树 菜单列表 |
| 2 | /api/DataDict/GetPageListByCondition | get | string condition, string keyWord,  string flid,int page,int limit |  | 点击左侧树节点 获取右侧数据字典列表  初始化时为condition:DataType,  keyword:type  flid 不用赋值  当点击后 condition:DataType,  keyword不用赋值  flid 为节点获取的DataTyp值 |
| 3 | /api/DataDict/GetListByCondition | get | string condition, string keyWord, |  | 字典分类的获取  condition：DataType  keyword:type |
| 2 | /api/DataDict/Create | post | model |  | Model 对应的key有 DataType ItemCode  ItemName Remark SortCode  如果新增的是根节点 DataType=type |
| 3 | /api/DataDict/Edit | post | model |  | Model 对应的key有 DataType ItemCode  ItemName Remark SortCode  如果修改的是根节点 DataType=type |
| 5 | /api/DataDict/Remove | post | string DataType, string ItemId |  | 如果删除的是根节点 DataType=type |
| 6 | /api/DataDict/RemoveBatch | post | List<string> DataType, List<string> ItemId |  | 如果删除的是根节点 DataType=type |

8.岗位管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Job/GetPageListByCondition | get | string orgid  string  depid condition, string keyword,int page,int limit |  | 获取岗位列表 带分页  orgid depid 单独传入  condition: JobName  keyword 为查询关键字 |
| 2 | /api/Job/Create | post | model |  | 与获取列表的字段相对应传入并加入OrgId OrgName DepId DepName |
|  | /api/Job/Edite | post | model |  | 与获取列表的字段相对应传入并加入OrgId OrgName DepId DepName 还要带入主键Id |
| 5 | /api/Job/Remove | post | keyValue |  | 主键 |
| 6 | /api/Job/RemoveBatch | post | keyValues |  | 主键数组 |

岗位成员分配

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /api/Job/GetJobUserList | get | string jobid,string orgId, int page, int limit |  | 获取所有用户信息 并把传入的有关jobid(岗位id主键)该岗位下已分配的用户加入标识为  LAY\_CHECKED = true; |
| 2 | /api/Job/Authorized | post | model |  | model为json字符串 key为  JobId UserId（选中的） |
| 3 | /api/Job/GetJobListById | get | String jobid |  | 根据jobid获取该岗位下的用户列表 |

# 三． 风险分级管理模块接口

1.风险单元管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/RiskUnitManage/GetPageListByCondition | get | string orgid  string condition, string keyword,int page,int limit |  | 获取风险单元列表 带分页  orgid  condition: OneName/TwoName/ThreeName  keyword 为查询关键字 |
| 2 | /bussiness/RiskUnitManage/Create | post | model |  | 与获取列表的字段相对应传入 注意 UntiBH UnitName为前端生成 格式如1/1/1采矿/露天采场生产系统/开拓运输 |
| 3 | /bussiness/RiskUnitManage/Edite | post | model |  | 与获取列表的字段相对应传入 并加入主键Id |
| 4 | /bussiness/RiskUnitManage/Remove | post | keyValue |  | 主键 |
| 5 | /bussiness/RiskUnitManage/RemoveBatch | post | keyValues |  | 主键数组 |

## **2**．风险点管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/RiskPointManage/GetPageListByCondition | get | String orgid  string condition, string keyword,int page,int limit |  | 获取风险点列表 带分页  orgid  condition:BH/Name/ZRR  keyword 为查询关键字 |
| 2 | /bussiness/RiskPointManage/Create | post | model |  | 与获取列表的字段相对应传入 |
| 3 | /bussiness/RiskPointManage/Edite | post | model |  | 与获取列表的字段相对应传入 并加入主键Id |
| 4 | /bussiness/RiskPointManage/Remove | post | keyValue |  | 主键 |
| 5 | /bussiness/RiskPointManage/RemoveBatch | post | keyValues |  | 主键数组 |

备注：这里的新建页面应带入机构下拉 风险单元下拉和对应的风险单元名称 以及风险点下拉 新增的时候风险点编号是前端自动生成的 格式如1/1/1-1 前面的是风险单元序号 后面的-1 是下拉获取的风险点序号 责任人 责任单位下拉

3.风险分级管控

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/RiskClassIfication/GetPageListByCondition | get | String orgid  string condition, string keyword,int page,int limit |  | 获取风险点分级列表 带分页  orgid  condition: RiskPointName/RiskLevel  keyword 为查询关键字 |
| 2 | /bussiness/RiskClassIfication/Create | post | model |  | 与获取列表的字段相对应传入 |
| 3 | /bussiness/RiskClassIfication/Edite | post | model |  | 与获取列表的字段相对应传入 并加入主键Id |
| 4 | /bussiness/RiskClassIfication/Remove | post | keyValue |  | 主键 |
| 5 | /bussiness/RiskClassIfication/RemoveBatch | post | keyValues |  | 主键数组 |

备注 新增的时候 R值不用传入

## 4.风险点清单

## 5.重大安全风险告知栏

# 四．岗位风险告知卡

1.岗位风险点管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/JobRiskManage/GetPageListByCondition | get | string condition, string keyword,int page,int limit |  | 获取列表 带分页  condition: OrgName/DepName/RoleName  RiskName/UserName  keyword 为查询关键字 |
| 2 | /bussiness/JobRiskManage/Create | post | model |  | 与获取列表的字段相对应传入 |
| 3 | /bussiness/JobRiskManage/Edite | post | model |  | 与获取列表的字段相对应传入 并加入主键Id |
| 4 | /bussiness/JobRiskManage/Remove | post | keyValue |  | 主键 |
| 5 | /bussiness/JobRiskManage/RemoveBatch | post | keyValues |  | 主键数组 |
| 6 | /business/JobRiskManage/GetUserListByJobId | get | jobid |  | 根据传入的jobid 获取该岗位下的所有人员信息 |
| 7. | /business/JobRiskManage/GetUserListByRoleId | get | roleid |  | 根据传入的roleid 获取角色下的所有人员信息 |

备注： 新增下拉顺序 机构->部门->角色

如果角色选取岗位员工 后面跟下拉岗位 –>人员列表 (下拉的接口上面都有)

如果选取其他角色 后面直接跟人员列表 不明白问我

风险点为该机构下的所有风险点下拉

## 2.岗位人员管理

岗位管理挪到这里

## 3员工岗位告知卡

# 五．安全检查管理

1.检查表管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/CheckTable/GetPageListByCondition | get | string condition, string keyword,int page,int limit |  | 获取列表 带分页  condition: RiskPointName/RiskLevel  keyword 为查询关键字 |
| 2 | /bussiness/CheckTable/GetClassClassIficationList | post | int page,int limit |  | 新增时 获取风险点分级管理列表 |
| 3 | /bussiness/CheckTable/Create | post | classificationId |  | 选中的风险点id主键数组 |
| 4 | /bussiness/CheckTable/RemoveBatch | post | keyValues |  | 主键数组 |

## 2.检查计划制定

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 路径 | 类型 | 输入 |  | 说明 |
| 1 | /bussiness/CheckPlan/GetPageListByCondition | get | string condition, string keyword, string starttime, string endtime, int page, int limit |  | 获取计划列表 带分页  condition: PlanName  keyword 为查询关键字  时间用starttime  endtime |
| 2 | /bussiness/CheckPlan /GetCheckTableListByUserid | get | string userid |  | 获取该人员下的检查点信息 |
| 3 | /bussiness/CheckPlan/Create | post | model |  | 对照着填写 model里不传入的为Id CreateTime LastCompleteTime RiskBH RiskName 多条用英文逗号分隔 |
| 4 | /bussiness/CheckPlan/RemoveBatch | post | keyValues |  | 主键数组 |

注释：新增计划时

做成下拉联动 机构->部门->角色->岗位->员工->该员工下的检查表信息

如果角色选的不是岗位员工

机构->部门->角色-->员工->该员工下的检查表信息