# 后台

### config/default.json（配置文件）

* db-host数据库地址和用户名密码
* app-port该后台应用的启动时的端口号

### node\_modules（项目所有依赖包）

### Equipment\_schema.js（定义成套设备表的数据结构）

* projectCode: String,（项目号，字符串类型）
* projectOrderCode: String,（项目顺序号，字符串类型）
* projectCompany: String,（项目所属公司，字符串类型）
* projectType: String,（项目类型，字符串类型）
* equipmentCode: String,（设备代号，字符串类型）
* equipmentName: String,（设备名称，字符串类型）
* equipmentGG: String,（设备规格，字符串类型）
* unitName: String,（单位名称，字符串类型）
* unitCode: String,（单位代号，字符串类型）
* resultCode: String,（最终代码，字符串类型）
* createTime: String,（创建时间，字符串类型）
* updateTime: String,（修改时间，字符串类型）
* deprecatedTime: String,（废弃时间，字符串类型）
* createUser: String,（创建人，字符串类型）
* updateUser: String,（修改人，字符串类型）
* deprecatedUser: String,（废弃人，字符串类型）
* deprecated: String,（是否废弃，字符串类型）

### index.js（后台所有接口）

* /userValidRepeat：接收workId（工号）作为参数，查询数据库中，如果结果长度大于0，则表示工号已存在
* /userAdd：接收workId（工号）、password（密码）、username（姓名）、rating（权限等级）、registerFlag（注册是否通过）、createTime（创建时间）作为参数，将该条数据存入用户表
* /userLogin：接收workId（工号）、password（密码）、registerFlag（注册是否通过）作为参数，判断输入的用户名和密码在数据库中是否存在，如果存在，且注册已经通过，则让该用户登录
* /userGetInfo：接收workId（工号）、registerFlag（注册是否通过）作为参数，查询出该工号的整个用户信息
* /userGetAllUnapproved：接收registerFlag（注册是否通过）作为参数，查询出所有注册未通过的用户
* /userApprovedOK：接收\_id（数据库中对应\_id）、registerFlag（注册是否通过）、rating（权限等级）作为参数，通过\_id查询到对应数据行，修改用户的注册状态为通过，并赋予一个权限等级
* /userUpdateRating：接收\_id（数据库中对应\_id）、rating（权限等级）作为参数，通过\_id查询到对应数据行，修改该行用户的权限等级
* /userValidOldPassword：接收workId（工号）、password（旧密码）作为参数，通过workId查询到指定用户，然后判断旧密码是否正确，校验成功则允许修改密码
* /userUpdateOldPassword：接收workId（工号），password（新密码）作为参数，通过workId查询到指定用户，然后使用新密码覆盖旧密码
* /unitGetAllData：不传入任何参数，直接查询全表数据，获取所有单位
* /materialGetCategoryByLevel：接收level（类别的级别）、deprecated（是否废弃）作为参数，查询指定类别级别下所有没有废弃的类别
* /materialGetCategoryById：接收parentId（父类id）作为参数，查询出所有parentId为该父类id的类别
* /materialGetRemoveRepeatXH：接收 thinId（细类id）作为参数，查询出所有该细类id的物料数据，然后根据xhms去重，将去重后的结果升序排序，然后返回给前端的列表展示
* /materialGetGGByXH：接收thinId（细类id）、xhCode（型号代号）、deprecated（是否废弃）作为参数，根据细类id加型号代号查询出所有规格，然后只将ggms（规格）、ggCode（规格代号）、level（类别的级别）返回给前端
* /materialGetTableAllData：接收deprecated（是否废弃）作为参数，获取所有未废弃的物料数据
* /materialGetTableDataByCategoryId：接收level（类别级别）、bigId（大类id）、middleId（中类id）、smallId（小类id）、thinId（细类id）、xhCode（型号代号）、ggCode（规格代号）作为参数，根据类别等级执行不同的操作，如类别级别是6级时，通过细类id加型号代号加规格代号查询出指定单条物料数据，如类别级别时5级时，通过细类id加上型号代号查询出所有规格的物料数据
* /materialGetLastCategoryId：不接受任何参数，获取类别表中最后一个类别的id
* /materialAddCategory：接收categoryName（类别名称）、categoryCode（类别代号）、level（类别等级）、id（类别id）、parentId（父类id）、createUser（创建人）作为参数，用于新增一个新类别
* /materialUpdateCategory：接收categoryName（类别名称）、username（用户名称）作为参数，修改类别名称
* /materialUpdateTable：接收level（类别级别）、id（类别id）、categoryName（类别名称）、username（用户名）、thinId（细类id）、xhCode（型号代号）、ggCode（规格代号）、updateUser（修改人）作为参数，根据不同的类别级别执行不同的操作，如在类别级别为3时，根据类别id查询出所有该id下的物料，然后修改其类别名称、修改人、修改时间
* /materialAddOne：接收bigName（大类名称）、bigCode（大类代号）、bigId（大类id）、bigParentId（大类的父id）、middleName（中类名称）、middleCode（中类代号）、middleId（中类id）、middleParentId（中类父id）、smallName（小类名称）、smallCode（小类代号）、smallId（小类id）、smallParentId（小类父id）、thinName（细类名称）、thinCode（细类代号）、thinId（细类id）、thinParentId（细类父id）、xhms（型号描述）xhCode（型号代号）、ggms（规格描述）、ggCode（规格代号）、unitName（单位名称）、unitCode（单位代号）、brandName（品牌名称）、createUser（创建人）作为参数，向物料表中写入一条数据
* /materialUpdateOne：接收\_id（数据库行\_id）、unitName（单位名称）、unitCode（单位代号）、brandName（品牌名称）、updateUser（修改人）作为参数，覆盖原有数据
* /materialGlobalSearch：此方法过于复杂，不再赘述
* /materialDeprecatedCategory：传入id（类别id）、level（类别级别）作为参数，根据类别id找到指定类别，然后将其标记为废弃
* /equipmentGetTableAllData：不传入任何参数，废弃所有未废弃的成套设备
* /equipmentAddOne：接收projectCode（项目号）、projectOrderCode（项目顺序号）、projectCompany（项目所属公司）、projectType（项目类别）、equipmentCode（设备代号）、equipmentName（设备名称）、equipmentGG（设备规格）、unitName（单位名称）、unitCode（单位代号）、createUser（创建人）作为参数，添加单个成套设备
* /equipmentUpdateOne：接收\_id(数据库行\_id)、equipmentName（设备名称）、equipmentGG（设备规格）、updateUser（修改人），修改单个成套设备
* /equipmentDeprecatedOne：resultCode（最终代码）、deprecatedUser（废弃人），废弃单个成套设备
* /homemadeGetTableAllData：不传入任何参数，查询出所有未废弃的自制件
* /homemadeAddOne：接收projectCode（项目号）、projectCompany（项目所属公司）、projectType（项目类别）、totalName（总装图名称）、totalCode（总装图代号）、partName（零部件名称）、partCode（零部件代号）、unitName（单位名称）、unitCode（单位代号）、resultCode（最终代号）、createUser（创建人）作为参数，添加单个自制件数据
* /homemadeUpdateOne：传入\_id（数据库行\_id）、partName（零部件代号）、updateUser（修改人）作为参数，修改单个自制件的部件名称并替换修改人
* /homemadeDeprecatedOne：传入\_id（数据库行\_id）、deprecatedUser（废弃人）作为参数，废弃单个自制件
* /operationHistoryAddOne：传入operationType（操作类型）、operationForm（操作的数据库表）、operationUser（操作人）、deprecatedData（废弃的数据）、updateBeforeData（修改前的数据）、updateAfterData（修改后的数据）、queryParm（封装了查询参数的对象）作为参数，根据不同的操作类型来执行不同的逻辑，如果操作类型为修改，则直接向操作表中写入一条操作记录，如果操作类型为删除，则先删除该类别下所有的子类，然后在废弃该类别下所有的编码，最后再写入一条删除记录，并将所有删除的数据记录到deprecatedData属性