

HRA人因数据采集系统 PAD端

系统简介

系统基于uniapp平台架构开发,打包成app后,暂只支持android平台. 由于该应用需要在无互联网无局域网的情况下离线使用, 采集的数据需要保存在pad上, 因此传统模式下的前后端交互将变为 数据的本地存储(SQLITE) + 数据同步(采集完成,需将数据同步到HRA PC端) 的模式.

数据库设计(数据模型)

1.本项目的数据存储方式为本地存储, 也就是数据存储在pad终端上,并不会直接存储到传统的后端服务器, 数据采集完成后,是需要用户手动将pad端的数据导入到PC端的.具体pad与pc端软件的数据同步的动作,不在该处讲述, 后续有章节具体说明.

2.本项目将采用SQLITE作为数据存储工具, 本地数据库与表都为动态创建,也就是在开发端只会保存该数据库的模型配置,在用户安装app后,首次登陆使用到sqlite后,才动态根据配置,动态创建数据库与表.

3.具体数据模型,请参看附件中的 model.svg

数据表

- **TUser** 用户表

备注: 登录页(pages/index/login.vue) 将从该表验证账号与密码

数据来源: 该表数据将从PC端接入,pad端不会插入新数据

数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型: service/user.js

- **TProject** 项目表

备注: 项目选择页(pages/index/index.vue)将从该表读取数据并显示

数据来源: 该表数据将从PC端接入,pad端不会插入新数据

数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型: service/project.js

- **TCatalog** 文件体系数据表

备注: 文件体系模块使用

leaf 用于区分文件夹与文件

parent_id 用于实现父子级

file 用于关联pad终端下的具体文件(只保存路径)

数据来源: PC数据同步与pad端新增

数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型: service/filesys/catalog.js

- **TFile** 文件体系数据表

备注: 用于全系统关联多媒体文件, 多媒体文件将保存在/storage/emulated/0/HRA_DOC/项目编

码/media/文件夹下, 每个项目单独会有一个统一文件夹存放资源 **type** 区分多媒体文件类型(1:图片,2:视频,3:语音)

foreign_id 用于该多媒体数据关联其他业务表的主键guid

name 保存其相关联业务表的表名称

field 保存其相关的业务表字段名称

path 用于保存多媒体文件在pad端的实际路径

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/filesys/file.js

- **TGroup** 人员配置分组表

备注 : 默认需要有 日常人员配置 | 运行班组 | 应急人员 三个组,可自定义新组

custom 区分类型 0 默认组 1 自定义组, 自定义的才可以修改组名与删除组

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/person/group.js

- **TDept** 人员配置 部门或科室

备注 : 人员组下的细分类别(一值,二值,三值)

group_guid 关联分组的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/person/dept.js

- **TJob** 岗位表

备注 : 岗位表

dept_guid 关联TDept表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/person/job.js

- **TPerson** 人员表

备注 : 岗位下的人员表

job_guid TJob 的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/person/person.js

- **Task** HRA 任务数据表

备注 : HRA任务列表

htype : 人因类别分为 A类 A类型相关性 C类 C类相关性

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/task.js

- **TypeA** HRA任务 A类主数据表 **task_id** : 关联Task表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/typeA.js

- **TYPEA_INFO** HRA任务 A类信息采集数据表

备注 : A类-信息采集 公共表(设备信息,操作对象,报警,指示,工作计划,试验验证,巡检,培训)

foreign_id : 关联TypeA表的guid

info_type :区分信息类型 device:设备信息, object:操作对象, warning:报警信息,directive 指示, plain:工作计划, verify:验证,training:培训

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/typeAInfo.js

- **inspection** A类信息采集 日常巡检
备注 : 属于A类数据的信息采集子表
foreign_id : 关联TypeA表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/accessibility.js
- **accessibility** A类信息采集 可达性数据表
备注 : 属于A类数据的信息采集子表
foreign_id : 关联TypeA表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/accessibility.js
- **Warning** A类信息采集 报警信息表 与C类信息采集公用
备注 : 属于A类数据的信息采集子表 (同用于C类信息采集表子表)
foreign_id : 关联TypeA表的guid (或关联TypeC数据的guid)
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/warning.js
- **deirective** A类信息采集 指示信息表 与C类信息采集公用
备注 : 属于A类数据的信息采集子表 (同用于C类信息采集表子表)
foreign_id : 关联TypeA表的guid (或关联TypeC数据的guid)
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/directive.js
- **plain** A类信息采集 计划信息表 **备注** : 属于A类数据的信息采集子表
foreign_id : 关联TypeA表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/plain.js
- **typeA_C** A类信息相关性 主数据表
备注 : 数据按组创建,一个组对应多条数据,列表中按组显示(显示组的第一条数据)
stage : 事件阶数(2阶对应两条,3阶对应3条,余此类推)
stage_index : 当前数据阶数
group_id : 记录当前数据组第一阶的guid
task_id : 关联Task表的guid
数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/typeAC.js

- **ASEP** ASEP事件树 **备注** : A类相关性 ASEP事件树表
foreign_id : 关联Type表的typeA_C或typeC_C 的guid
related_text : 相关性文本值(低相关 高相关 中相关 等等)
related_value : 相关性数值(1,2,3,4等) 具体值代表是第几条路径,用于UI绘图路径显示
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/asep.js
- **THERP** THERP事件树 **备注** : A类相关性,c类相关性共用的 THERP事件树表
foreign_id : 关联Type表的typeA_C或typeC_C 的guid
related_text : 相关性文本值(低相关 高相关 中相关 等等)
related_value : 相关性数值(1,2,3,4等) 具体值代表是第几条路径,用于UI绘图路径显示
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/therp.js
- **TYPE_C** HRA任务 C类主数据表 **task_id** : 关联Task表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/typeC.js
- **TYPE_C_INFO** HRA任务 C类信息采集数据表
备注 : HRA数据 C类-信息采集 公共表(报警,指示,主控班组,规程)
foreign_id : 关联TypeA表的guid
info_type : 区分信息类型 warning:报警信息,directive 指示, team 主控班组信息, procedure 规程
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/typeCInfo.js
- **behavior_type** C类信息采集->行为类型
备注 : 属于C类数据的信息采集子表
foreign_id : 关联TypeC表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/behaviorType.js
- **analysis** C类信息采集->人员分析需求信息(情景信息)
备注 : 属于C类数据的信息采集子表
foreign_id : 关联TypeC表的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/hra/analysis.js
- **fault_event** C类信息采集->事件树,故障树(情景信息)
备注 : 属于C类数据的信息采集子表
foreign_id : 关联TypeC表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/faultEvent.js

- **timing** C类信息采集->时序(情景信息)

备注 : 属于C类数据的信息采集子表

foreign_id : 关联TypeC表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/timing.js

- **PSF** C类信息采集->PSF

备注 : 属于C类数据的信息采集子表,psf数据分为诊断与执行两条数据,通过type区分

foreign_id : 关联TypeC表的guid

type : 数据类别: 1 诊断 2 执行 **数据来源** : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/psf.js

- **accessibility_c** C类信息采集->可达性

备注 : 属于C类数据的信息采集子表

foreign_id : 关联TypeA表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/accessibilityC.js

- **on_site_equipment** C类信息采集->就地设备可操作性

备注 : 属于C类数据的信息采集子表

foreign_id : 关联TypeC表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/onSiteEquipment.js

- **CBDT** C类信息采集->就地设备可操作性

备注 : 属于C类数据的信息采集子表

foreign_id : 关联TypeC表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/CBDT.js

- **typec_regulation** C类信息采集 -> 规程步骤

备注 : 属于C类数据的信息采集子表

foreign_id : 关联TypeC表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/typeCRegulation.js

- **typeC_C** C类信息相关性 主数据表

备注 : 数据按组创建,一个组对应多条数据,列表中按组显示(显示组的第一条数据)

stage : 事件阶数(2阶对应两条,3阶对应3条,余此类推)

stage_index : 当前数据阶数

group_id : 记录当前数据组第一阶的guid

task_id : 关联Task表的guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/typeCC.js

- **nureg1921** nureg1921事件树 **备注** : c类相关性 nureg1921事件树

foreign_id : 关联Type表的typeA_C或typeC_C 的guid

related_text : 相关性文本值(低相关 高相关 中相关 等等)

related_value : 相关性数值(1,2,3,4等) 具体值代表是第几条路径,用于UI绘图路径显示

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/nureg1921.js
- **SPAR_H** HRA SPAR_H 事件树 **备注** : c类相关性 SPAR_H

foreign_id : 关联Type表的typeA_C或typeC_C 的guid

related_text : 相关性文本值(低相关 高相关 中相关 等等)

related_value : 相关性数值(1,2,3,4等) 具体值代表是第几条路径,用于UI绘图路径显示

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/hra/sparH.js
- **device** 问题普查=>设备表

备注 : 问题普查将基于设备进行多个方面的问题调查

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/investigation/device.js
- **investgation** 问题普查=> 设备 => 基于多方面的问题普查结果表

备注 : 问题普查将基于设备进行多个方面的问题调查

foreign_id : 关联设备表device 的 guid

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/investigation/investgation.js
- **intervier_config** 访谈人员配置 信息表

备注 : 记录访谈人员 访谈对象 访谈地点

type : 数据类型 (访谈人员,访谈对象,访谈地点)

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/config/intervier_config.js
- **situation** 情境信息表(ISV)

备注 : ISV -> 情境信息表

数据来源 : PC数据同步与pad端新增

数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据

关联模型 : service/isv/situation.js

- **situation_version** 情境版本表(ISV)
备注 : ISV -> 情境信息表->情境版本表
foreign_id : 关联情境信息表situation 的 guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/isv/situationVersion.js
- **situation_step** 情境步骤表(ISV)
备注 : ISV -> 情境信息表->情境版本表 -> 情境步骤
foreign_id : 关联情境版本situation_version 的 guid
step_guid : 关联基础数据中规程的步骤guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/isv/situationStep.js
- **situation_issue** 情境问题表(ISV)
备注 : ISV -> 情境信息表->情境版本表 -> 情境步骤 -> 情境问题
foreign_id : 关联情境步骤situation_step 的 guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/isv/situationIssue.js
- **situation_score** 评分表(ISV)
备注 : ISV -> 情境信息表-> 评分
foreign_id : 关联情境信息表situation 的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/isv/situationScore.js
- **plant** 电厂基础信息表(人员效能)
备注 : 人员效能 -> 电厂基础信息
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/effect/plant.js
- **scene** 场景信息表(人员效能)
备注 : 人员效能 -> 电厂基础信息 -> 场景
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/effect/scene.js
- **scene_item** 失效/屏蔽 设备表(人员效能)
备注 : 人员效能 -> 电厂基础信息 -> 场景编辑 的部件
foreign_id : 关联场景信息表scene 的guid
数据来源 : PC数据同步与pad端新增
数据接入方式 : 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型 : service/effect/sceneItem.js

- **toe** 情境因子(人员效能)
备注: 人员效能 -> 综合数据采集 -> 情境因子
数据来源: PC数据同步与pad端新增
数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型: service/effect/toe.js
- **performance** 评价表(人员效能)
备注: 情境步骤的评价 操作流程步骤评价 综合数据采集评价 共用表
foreign_id: 关联 场景操作步骤 | isv情境步骤 | 场景的guid
数据来源: PC数据同步与pad端新增
数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型: service/effect/performance.js
- **lapse** 失误模式(人员效能)
备注: 与performance绑定在一起
foreign_id: 关联 场景操作步骤 | isv情境步骤 | 场景的guid
数据来源: PC数据同步与pad端新增
数据接入方式: 初始化项目的时候,读取从PC端copy的proj.json文件获取数据
关联模型: service/effect/lapse.js

数据模型

数据模型用于操作数据表,比如对表插入数据,查询数据,修改数据乃至删除数据.

每个表都对应有一个数据模型. 数据模型包含对应表的各种业务操作. 通过抽象数据模型的公共逻辑代码, app中实现了service/base 模型基类. BaseService 封装了对SQLite 数据表的各种操作方法,基本能满足各种场景. 其他数据模型都将继承BaseService 而无需自己实现对表的CRUD方法. 除非具有特个性的操作,需要自己实现或覆盖父类方法.

子类Service 只需要定义数据表的DDL

举例说明:人员效能-场景信息数据模型, 该模型字需要配置scene表的ddl定义,然后就可以通过继承BaseService提供的方法,对表进行各种CRUD操作

```
import constants from '../common/constants'
import BaseService from '../base'
import toeService from './toe'
//情境信息表
let config = {
  tableName: "scene",
  columns: [
    { name: "guid", type: "VARCHAR(50)", ext: "PRIMARY KEY" }, //唯一ID
    { name: "foreign_id", type: "VARCHAR(50)", ext: "NOT NULL" }, //关联的主数
据ID 关联基础信息
    { name: "name", type: "VARCHAR(50)", ext: "NOT NULL" }, //场景/事件/概况
    { name: "start_time", type: "VARCHAR(50)" }, //初始事件引出时间
    { name: "end_time", type: "VARCHAR(50)" }, //事件结束时间
    { name: "team_name", type: "VARCHAR(50)" }, // 班组名称
    { name: "project_id", type: "VARCHAR(50)" }, //所属项目ID
    { name: "created_at", type: "INT" }, //任务创建时间
```



```
    { name: "created_by", type: "VARCHAR(50)" }, //创建人
    { name: "updated_at", type: "INT" }, //修改人
    { name: "updated_by", type: "VARCHAR(50)" }, //修改时间
    { name: "delete_flag", type: "INT", ext: " DEFAULT 0 " }
  ],
}
class Service extends BaseService {

}
const service = new Service(config)
export default service;
```

应用代码目录说明

- **_doc** : 内部私有目录,存放SQLITE数据库文件
- **common** : 公共的js与css文件目录
- **components** : 公共组件目录(包含外部导入的与app自定义的组件)
- **datasync** : app与外部系统数据同步使用的数据解析模块
- **pages** : 应用UI页面总目录,内部按业务模块分类存放
- **service** : 本地存储模型目录,负责sqlite数据建模与本地数据存储业务
- **static** : 静态资源目录,存放字体图标与图片资源
- **store** : vuex 状态共享文件目录

功能模块

登录

相关UI页面: **pages/index/login.vue**

相关模型文件: **service/user.js**

数据表名称: **TUser** (数据来源于与后端数据同步,pad不插入新数据)

功能说明:

- 离线登录,查询数据表,匹配用户名与密码
- 缓存用户上次登录用户名与密码,下次一键登录
- 自动登录 如果用户不退出,则始终保存登录状态

首页(项目选择页)

相关UI页面: **pages/index/index.vue**

相关模型文件: **service/project.js**

数据表名称: **TProject**

功能说明:

- 展示项目数据
- 进入项目,如果项目未初始化过,需要初始化基础数据

service/project.js 项目模型配置(用于动态创建数据表,初始化数据) project.js

```
let config = {
  tableName: "TProject", //表名称
  //列配置
  columns: [
    { name: "guid", type: "VARCHAR(50)", ext: "PRIMARY KEY" },
    { name: "projectid", type: "INT" },
    { name: "code", type: "VARCHAR(50)" },
    { name: "name", type: "VARCHAR(50)", ext: "NOT NULL" },
    { name: "init", type: "INT" }, //是否已经初始化(同步过pc数据)
    { name: "init_time", type: "INT" } //用于记录该项目上次初始化的时间(同步PC端数据数据)
  ],
  //初始化数据(正式环境不应该有该配置,该处是为了测试添加初始数据)
  datas: [
    ['JIE49483KFJD223', 100, 'PROJECT_001', '福清核电站', 0, ''],
    ['JIE49483KFJD224', 101, 'PROJECT_002', '田湾核电站', 0, ''],
    ['JIE49483KFJD225', 102, 'PROJECT_003', '徐大堡核电站', 0, '']
  ],
  // rebuild: true
}
```

文件体系

相关UI页面: **pages/filesys/**

相关模型文件: **service/filesys/**

相关数据表: **TCatalog , TFile**

功能说明:文件体系目录与文件管理功能, 文件可关联pad下的文件, 并可以添加多媒体文件(图片,视频,音频)

人员配置

相关UI页面: **pages/person/**

相关模型文件: **service/person/**

相关数据表: **TGroup , TDept, TJob, TPerson**

功能说明:文件体系目录与文件管理功能, 文件可关联pad下的文件, 并可以添加多媒体文件(图片,视频,音频)

HRA数据

相关UI页面: **pages/hra/**

相关模型文件: **service/hra/**

相关数据表: **Task , TypeA , TYPEA_INFO,inspection,accessibility ,Warning,deirective,plain,typeA_C,ASEP,THERP,TYPE_C,behavior_type,analysis,fault_event,timing,PSF,accessibility_c,on_site_equipment,CBDT,typec_regulation,typeC_C,nureg1921,SPAR_H**

功能说明:hra任务数据采集 包含A类 C类 A类相关性 C类相关性

ISV

相关UI页面: **pages/isv/**

相关模型文件: **service/isv/**

相关数据表: **situation , situation_version, situation_step, situation_issue, situation_score**

功能说明:ISV 包含 情境信息,情境版本 情境步骤,情境评分等功能

人员效能

相关UI页面: **pages/effect/**

相关模型文件: **service/effect/**

相关数据表: **plant , scene, scene_item, toe, performance, lapse**

功能说明:ISV 包含 情境信息,情境版本 情境步骤,情境评分等功能

问题普查

相关UI页面: **pages/investigation/**

相关模型文件: **service/investigation/**

相关数据表: **device , investgation**

功能说明:设备的问题普查

访谈人员配置

相关UI页面: ****pages/index/config.vue**

相关模型文件: ****service/config/Interview_config.js**

相关数据表: **intervier_config**

功能说明:访谈人员配置包含 访谈人员,访谈对象,访谈地址三类型

我的

相关UI页面: ****pages/index/mine.vue**

相关模型文件: 无

相关数据表: 无

功能说明: 切换项目| 退出登录 | 数据下载(同步基础数据)

自定义组件

comp-media 多媒体组件

关键方法: popup(foreign_id, field, tableName).

组件说明:用于在任何表单或页面上关联多媒体文件(图片,视频,语音).

html 引入组件:

```
<comp-media ref="media"></comp-media>
```

javascript 调用方法,打开多媒体文件管理器:

```
this.$refs.media.popup(this.guid, 'remark', 'CBDT');
```

comp-page 表单组件

组件说明:可根据数据模型(service) 生成表单 , 需要service实现**getFormItems**方法 html 引入组件:

```
<comp-page
  ref="compPage"
  :service="service"
  title=""
  :param="loadParam"
  :autoload="true"
  :header="false"
  :scroll="false"
  formcss="step-form"
></comp-page>
```

comp-related 事件相关性选择组件

组件说明:用于显示并选择事件相关性. 需要service实现 **getRelatedFields** 方法,该方法返回相关性的选项名称,另外还需要实现 **getRelatedTree** 方法,该方法返回 相关性选择的分支详情. html 引入组件:

```
<comp-related
  ref="compRelated"
  :service="service"
  v-if="service"
  :value="value"
  @change="onFlowChange"
  :fix_right="fixRight"
></comp-related>
```

comp-table 表格组件

组件说明:用于hra 信息采集中的表格展示(报警,指示,计划等).并支持打开弹出form,新增数据.需要service 实现 **getTableItems** 方法,该方法需要返回表格的列配置.如需支持打开form,新增数据,需要实现**getFormItems**方法,返回表单配置

html 引入组件:

```
<comp-table
  v-if="tableService && tableParam"
  :service="tableService"
  :param="tableParam"
  :addTitle="addTitle"
>
</comp-table>
```

comp-tag 标签输入控件

组件说明:用于文件体系中的 文件类型, 文件类型可选择已有标签,也可以在动态添加新标签.

html 引入组件:

```
<comp-tag type="文件类别" @change="onTypeChange" :value="file.type">
</comp-tag>
```

uni-tab-bar

组件说明:自定义home tab标签,可支持底部横向,或左侧竖向显示.目前因为用户确认,暂锁定在底部显示

html 引入组件:

```
<uni-tab-bar :class="screenOrientation" index="6"></uni-tab-bar>
```

PAD从PC端获取基础数据(需与PC端对接,后续优化)

需要采用线下方式,将pc端生成的项目目录(以项目guid命名),复制到 **HRA_DOC/** 目录下,目录下需要包含 **proj.json** 数据文件,项目相关的多媒体文件目录meida, 还有其他相关文件. proj.json 读取动作是在**datasync.js**中的**initProject**方法中进行的,后续需要添加对该文件的数据读取,将数据插入 SQLite 相关数据表.

pad 初始化时候,会读取该文件夹proj.json 初始化项目. 另外还需 **HRA_DOC/global.json** 该json是项目数据文件,用于初始化项目列表,登录用户里列表,还有一些其他与具体项目无关的基础数据.

datasync.js 的 **initApp** 方法中实现(后续需要添加基础数据的存储动作)

PAD将采集数据上传到PC端

目前的方案,为线下上传. 请执行数据打包操作后,手动将pad端下的文件复制到PC端, 具体拷贝以项目guid命名的压缩文件 **HRA_DOC/xxx.zip/**

应用安装与使用

- 1.安装应用APK
- 2.在pad根目录下创建HRA_DOC目录
- 3.将PC提供的golabl.json文件拷贝到HRA_DOC 目录下! golabl.json中包含了项目列表与登录用户数据
- 4.将PC提供的项目文件夹复制到HRA_DOC 目录下. 项目文件夹包含了多媒体文件夹media, 项目数据文件proj.json.还可以有其他相关文件