**智慧水务-河湖管养数字化管理平台-工程管理建设项目**

**开发文档**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | V1.0. |
| 作 者： | 刘成、李灿、徐藤 |
| 完成日期： | 2023年8月18日 |

**版权信息**

**本文件的版权属于政浩软件（上海）有限公司**

**任何形式的散发都必须先得到政浩软件（上海）有限公司的许可**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **修订记录** | | | | | |
| 变更类型：A-新增，M-修改，D-删除 | | | | | |
| 版本 | 日期 | 变更类型 | 修订人 | 审核人 | 修订原因 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1 架构设计 1](#_Toc144814234)

[1.1 微服务架构 1](#_Toc144814235)

[1.2 Spring开发框架 1](#_Toc144814236)

[1.3 数据库技术 3](#_Toc144814237)

[1.4 空间信息技术 4](#_Toc144814238)

[2 概要设计 5](#_Toc144814239)

[2.1 工程 5](#_Toc144814240)

[2.2 台账信息设计概要 7](#_Toc144814241)

[3 详细设计 8](#_Toc144814242)

[3.1 代建管理和总包管理 8](#_Toc144814243)

[3.2 工程模板设计 8](#_Toc144814244)

[3.3 台账信息设计 9](#_Toc144814245)

[4 数据库设计 10](#_Toc144814246)

# 架构设计

## 微服务架构

Spring Cloud是一系列框架的[有序集合](https://baike.baidu.com/item/%E6%9C%89%E5%BA%8F%E9%9B%86%E5%90%88/994839?fromModule=lemma_inlink" \t "_blank)。它利用[Spring Boot](https://baike.baidu.com/item/Spring%20Boot/20249767?fromModule=lemma_inlink" \t "_blank)的开发[便利性](https://baike.baidu.com/item/%E4%BE%BF%E5%88%A9%E6%80%A7/53294221?fromModule=lemma_inlink" \t "_blank)巧妙地简化了[分布式系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F/4905336?fromModule=lemma_inlink" \t "_blank)基础设施的开发，如[服务发现](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%8F%91%E7%8E%B0/23734240?fromModule=lemma_inlink" \t "_blank)注册、配置中心、[消息总线](https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E6%80%BB%E7%BA%BF/58531073?fromModule=lemma_inlink" \t "_blank)、[负载均衡](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1/932451?fromModule=lemma_inlink" \t "_blank)、[断路器](https://baike.baidu.com/item/%E6%96%AD%E8%B7%AF%E5%99%A8/5921574?fromModule=lemma_inlink" \t "_blank)、[数据监控](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%9B%91%E6%8E%A7/16044113?fromModule=lemma_inlink" \t "_blank)等，都可以用Spring Boot的开发风格做到一键启动和部署。Spring Cloud并没有重复制造轮子，它只是将各家公司开发的比较成熟、经得起实际考验的[服务框架](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E6%A1%86%E6%9E%B6/1507180?fromModule=lemma_inlink" \t "_blank)组合起来，通过Spring Boot风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统[开发工具](https://baike.baidu.com/item/%E5%BC%80%E5%8F%91%E5%B7%A5%E5%85%B7/10464557?fromModule=lemma_inlink" \t "_blank)包

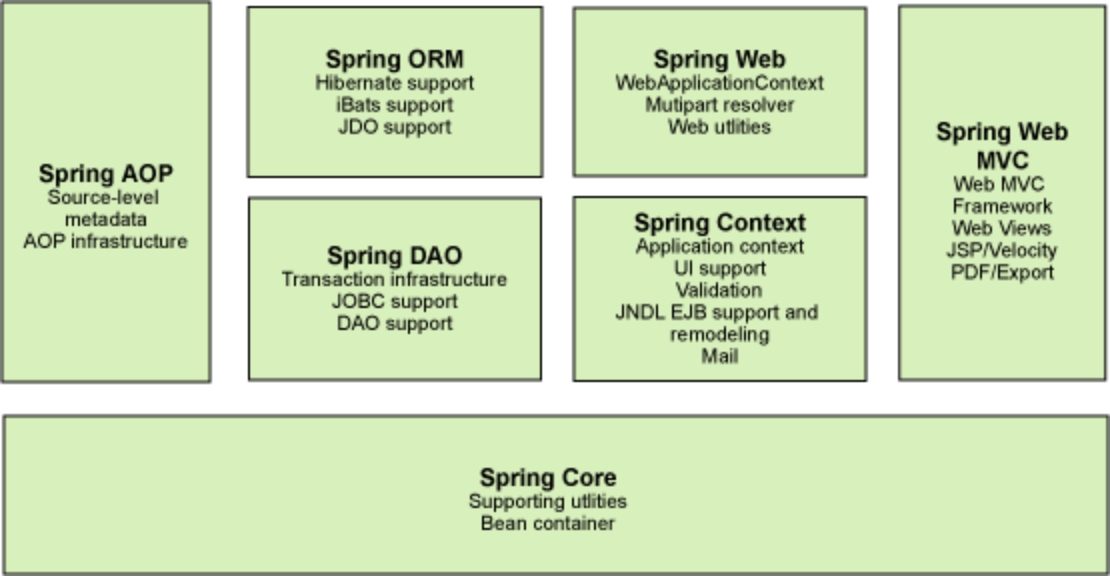
## Spring开发框架

采用Spring开发框架，该框架是一个分层架构，由7个定义良好的模块组成。Spring模块构建在核心容器之上，核心容器定义了创建、配置和管理bean的方式。

Spring 框架的功能可以用在任何J2EE服务器中，大多数功能也适用于不受管理环境。

Spring 的核心要点是：支持不绑定到特定J2EE服务的可重用业务和数据访问对象。毫无疑问，这样的对象可以在不同 J2EE 环境（Web 或 EJB）、独立应用程序、测试环境之间重用。

Spring是一个解决了许多在J2EE开发中常见的问题的强大框架。 Spring提供了管理业务对象的一致方法并且鼓励了注入对接口编程而不是对类编程的良好习惯。Spring的架构基础是基于使用JavaBean属性的Inversion of Control容器。



组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。每个模块的功能如下：

* **核心容器**：核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是工厂模式的实现。BeanFactory使用控制反转（IOC）模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。
* **Spring 上下文：**Spring 上下文是一个配置文件，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化、校验和调度功能。
* **Spring AOP：**通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。
* **Spring DAO：**JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。
* **Spring ORM：**Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。
* **Spring Web 模块：**Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。
* **Spring MVC 框架：**MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成为高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

Spring在使用IOC容器作为构建所有架构层的完整解决方案方面是独一无二的。

* Spring提供了唯一的数据访问抽象，包括简单和有效率的JDBC框架，极大的改进了效率并且减少了可能的错误。
* Spring的数据访问架构还集成了Hibernate和其他O/R mAPPing解决方案。Spring还提供了唯一的事务管理抽象，它能够在各种底层事务管理技术，例如JTA或者JDBC事务提供一个一致的编程模型。
* Spring提供了一个用标准Java语言编写的AOP框架，它给POJOs提供了声明式的事务管理和其他企业事务。如果需要，还能实现你自己的aspects。这个框架足够强大，使得应用程序能够抛开EJB的复杂性，同时享受着和传统EJB相关的关键服务。

Spring提供了可以和IOC容器集成的强大而灵活的MVC Web框架。

## 数据库技术

数字化城管系统的建设必然需要借助于数据库技术特别是数据库新技术。值得采用的数据库新技术包括：分布式数据库技术、数据仓库技术、面向对象数据库技术、多媒体数据库技术、Web数据库技术、第三级存储器、空间数据存储技术以及信息检索与浏览技术等。

## 空间信息技术

空间信息技术是以遥感技术（RS）、地理信息系统（GIS）、全球定位系统（GPS）为基础，将RS、GIS、GPS三种独立技术领域中的有关部分与其它高技术领域（如网络技术、通讯技术等）有机地构成一个整体而形成的一项新的综合技术。

（1）GIS技术

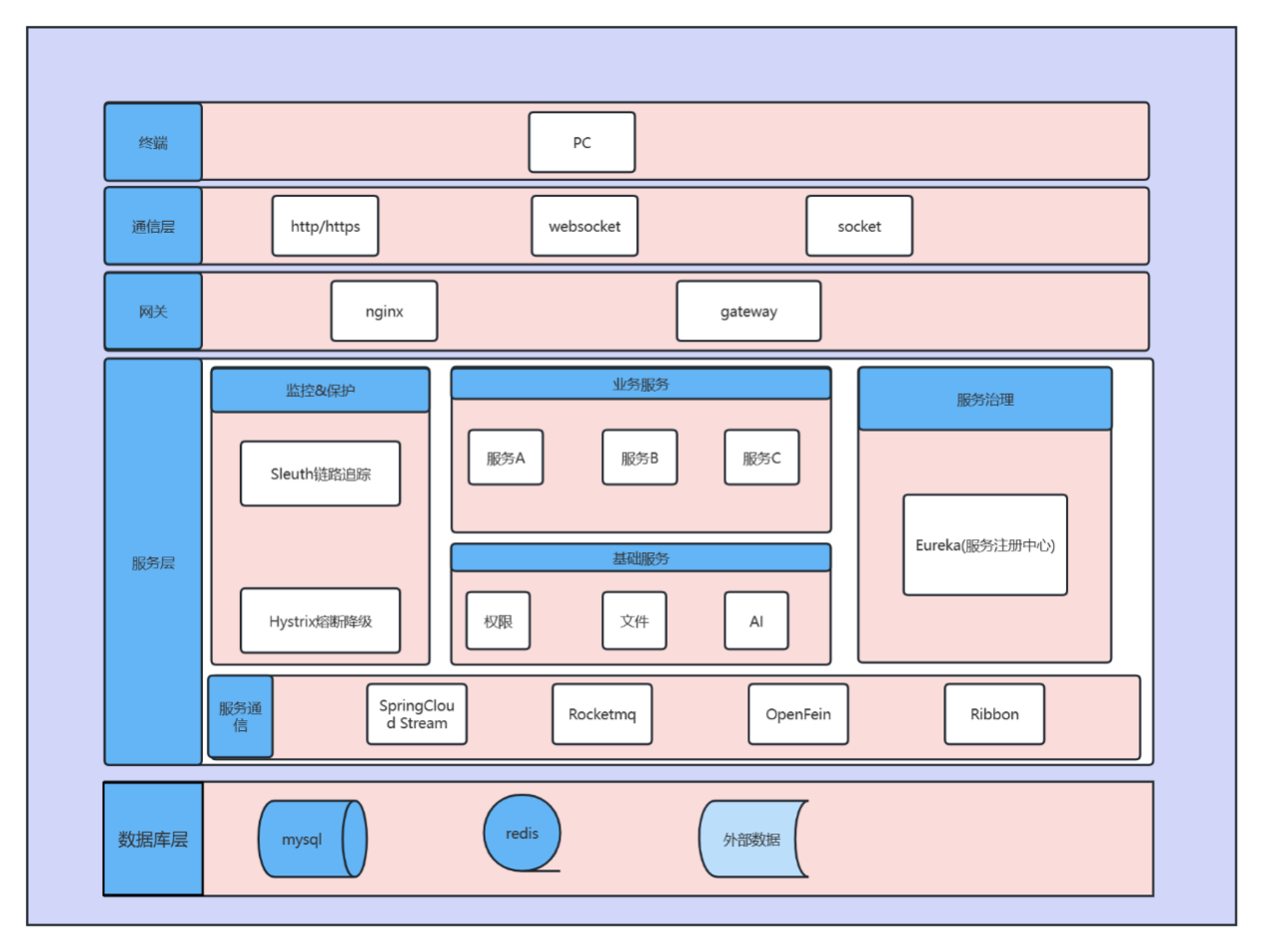
地理信息系统简称GIS（Geographical-Information-System)。地理信息系统是以地理空间数据库为基础，采用地理模型分析方法，适时提供多种空间和动态的地理信息，为地理研究和地理决策服务的计算机技术系统。

在本次智慧平台购买服务中，充分融合地理信息系统（GIS）技术，实现对基础地理数据、城市单元网格数据等地理空间数据的管理、维护、显示、操作、分析和建模等。实现整个系统管理和显示地理空间数据，图文一体化办公的基础平台。

（2）GPS技术

全球定位系统简称GPS（Global-Positioning-System），经我国测绘等部门的使用表明，GPS以全天候、高精度、自动化、高效益等显著特点，赢得广大测绘工作者的信赖，并成功地应用于大地测量、工程测量、航空摄影测量、运载工具导航和管制、地壳运动监测、工程变形监测、资源勘察、地球动力学等多种学科，从而给测绘领域带来一场深刻的技术革命。

在本次智慧平台购买服务中，GPS全面应用于人员、车辆监管等系统，借助于GPS技术并结合GIS技术，环卫固废平台可以方便地实现对人员、车辆的空间定位和可视化管理。



技术架构图

# 概要设计

## 工程

**工程数量统计：**

按照四种维度展示代建工程的工程数量：总数、施工前、施工中、施工后

**工程资金统计：**

展示所有代建工程的资金信息进行统计，分别按照工程总投资（合同）金额、累计支付总金额、累计未支付总金额、累计支付总比例、当年总预算金额、当年总预算执行率来进行统计。所有的取值，通过对工程台账管理中，获取数据

**各统计计算逻辑方式**

**工程总投资（合同）金额：**计算所有代建合同的项目总投资金额的合计

**累计支付总金额：**所有代建项目已支付总金额的合计。

**累计未支付总金额：**所有代建项目未支付总金额的合计。

**累计支付总比例：**所有代建项目累计支付总金额/所有代建项目总投资金额\*100%

**年总预算金额：**所有代建项目的合计预算金额，默认展示当年预算金额。不和上面的年份筛选做交互。

**年总预算执行率：**所有代建项目工程的支付金额/所有代建项目工程的项目预算金额\*100%

**工程资金统计筛选逻辑方式**

**请选择年份：**选择年份的下拉展示年份信息，按照代建工程立项日期来进行筛选。（选择年份不影响总预算金额、总预算执行率）

**请选择工程：**下拉查找所有代建工程的工程名称信息。

**请选择工程状态：**

选择了全部，可查看所有阶段中的资金信息

选择了进行中，可查看所有阶段在进行中的资金信息

选择了销项，只查看销项阶段的资金信息

**工程模板设计概要:**

工程模板主要由三张表组成

en\_engineering\_template\_main(工程模板主表) en\_engineering\_template\_stage\_info(工程模板阶段信息表)

en\_engineering\_template\_stage\_matters(工程模板阶段事项表).

关联关系为工程模板主表id关联工程模板阶段信息表main\_id字段在查询模板主表的某一条详情时候带出阶段信息，阶段信息又和事项信息关联 由阶段信息的id关联事项信息的stage\_id字段 每个阶段信息下面会有多个事项信息。所以在查询相互信息的时候可以根据关联id进行左连接查询数据。

## 台账信息设计概要

台账信息主要由以下几张表组成:

en\_engineering\_contract\_ledger\_main(合同台账信息主表)

en\_engineering\_contract\_ledger\_amount\_details(合同台账信息本期支付信息表)

en\_engineering\_agent\_construction(工程主表)

首先台账信息由工程项目决定 一个代建工程或者总包工程关联一个台账信息，由工程主表id关联合同台账信息主表engineering\_id字段进行关联。在查询台账列表的时候主要查询工程主表信息然后获取台账列表信息再关联台账主表信息进行保存编辑删除等功能。在台账信息里面 每一条台账对应多条支付信息，由台账信息主表id关联本期支付信息表main\_id字段，再通过查询台账信息id关联字段查询住支付信息返回并进行修改删除新增等操作。

# 详细设计

## 代建管理和总包管理

1.1：列表：查询代建主表，通过状态区分代建和总包

1.2：新增和修改：为同一接口，通过id区分。新增时增加资金表，主表，阶段表，图片表数据

1.3：删除：通过id更改数据删除状态，进行逻辑删除

1.4：代建详情接口：通过id查询详情

1.5：项目进度图：通过id查询详情，返回阶段数据

1.6：项目日志:通过id查询详情，通过主表id查询列表，新增基于主表id进行新增，修改基于id进行修改，删除基于id进行逻辑删除

1.7：操作日志：对工程所有接口进行监控，获取返回值和参数并记录下来返回给前端

1.8 ：en\_engineering\_agent\_construction 为主表，其他工程表全部关联与主表

## 工程模板设计

**工程模板列表接口**

Swagger地址http://47.102.116.59:8998/doc.html#/default/%E5%B7%A5%E7%A8%8B%E6%A8%A1%E6%9D%BFApi/getTemplateListUsingPOST\_3

传参为查询条件和分页参数 取en\_engineering\_template\_main表数据通过查询未删除，指定条件数据返回列表信息。

**工程模板阶段详情**

Swagger地址

http://47.102.116.59:8998/doc.html#/default/%E5%B7%A5%E7%A8%8B%E6%A8%A1%E6%9D%BFApi/getTemplateInfoUsingPOST

传参为模板主键id 通过en\_engineering\_template\_main和en\_engineering\_template\_stage\_info阶段表信息关联 在点击对应列表的信息时传主表id信息 然后关联阶段信息表查询阶段信息给前端。阶段详情并且可以排序 删除 修改。

**工程模板阶段事项**

Swagger地址

http://47.102.116.59:8998/doc.html#/default/%E5%B7%A5%E7%A8%8B%E6%A8%A1%E6%9D%BFApi/getTemplateInfoUsingPOST

通过en\_engineering\_template\_stage\_info和

en\_engineering\_template\_stage\_matters关联，在阶段详情下查询事项信息返回，并且可以删除 排序 修改 新增。

**工程模板保存接口**

Swagger地址

[http://47.102.116.59:8998/doc.html#/default/%E5%B7%A5%E7%A8%8B%E6%A8%A1%E6%9D%BFApi/saveTemplateInfoUsingPOST](http://47.102.116.59:8998/doc.html" \l "/default/%E5%B7%A5%E7%A8%8B%E6%A8%A1%E6%9D%BFApi/saveTemplateInfoUsingPOST)

传参为工程模板的实体类对象,保存修改新增都是同一个接口 根据id来区分有id为修改原数据无id为新增参数。

## 台账信息设计

台账信息分为代建台账和工程台账，其中只有小部分有却别所以在设计开始为台账设计成同一张表。通过台账信息主表en\_engineering\_contract\_ledger\_main中的type字段区分时代建工程台账还是总包工程台账 （台账类型0-代建工程 1-总包工程）。之后再新增，修改，查询的时候都通过传对应的type值来确定时总包工程还是代建工程的台账。

**台账信息列表**

Swagger地址

http://47.102.116.59:8998/doc.html#/default/%E5%8F%B0%E8%B4%A6Api/getTemplateListUsingPOST\_2

传参为查询条件和分页参数，台账信息列表时通过工程表去查询的，查询工程里面的信息，取出相对应的字段组合成列表返回给前端，查询可以分页查询支持条件搜索。

**台账信息详情查询**

Swagger地址

[http://47.102.116.59:8998/doc.html#/default/%E5%8F%B0%E8%B4%A6Api/getEngineeringContractLedgerInfoUsingPOST](http://47.102.116.59:8998/doc.html" \l "/default/%E5%8F%B0%E8%B4%A6Api/getEngineeringContractLedgerInfoUsingPOST)

传参是工程id

台账信息的查询是通过三张表进行的。分别是en\_engineering\_agent\_construction（工程表） en\_engineering\_contract\_ledger\_main（台账信息主表）

en\_engineering\_contract\_ledger\_amount\_details（台账收入金额表）

台账信息关联工程表id主教 收入金额表关联台账主表，进行关联查询。查询出信息之后对信息进行统计，计算出年度统计和月度统计返回给前端做展示。

**本期金额列表接口**

Swagger地址

[http://47.102.116.59:8998/doc.html#/default/%E5%8F%B0%E8%B4%A6Api/geContractLedgerAmountDetailsUsingPOST](http://47.102.116.59:8998/doc.html" \l "/default/%E5%8F%B0%E8%B4%A6Api/geContractLedgerAmountDetailsUsingPOST)

传参为台账id 根据台账id去en\_engineering\_contract\_ledger\_amount\_details（台账收入金额表）查询

然后返回当期参数封装

3.4 工程可视化大屏

工程阶段管理：

查询工程和对应阶段，并且通过状态和文件是否存入进行预警

网格专题：

查询全部的12个街道和对应的数量，并且能将工程位置展示在地图上

# 数据库设计

CREATE TABLE `en\_engineering\_construction` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT 'id',

`project\_name` varchar(255) DEFAULT NULL COMMENT '工程名称',

`project\_approval\_time` date DEFAULT NULL COMMENT '工程立项时间',

`application\_number` varchar(255) DEFAULT NULL COMMENT '工程报建编号',

`project\_leader` varchar(255) DEFAULT NULL COMMENT '工程项目负责人',

`commencement\_time` date DEFAULT NULL COMMENT '工程开工时间',

`completion\_time` date DEFAULT NULL COMMENT '工程完工时间',

`jungong\_time` date DEFAULT NULL COMMENT '工程竣工时间',

`construc\_unit` varchar(255) DEFAULT NULL COMMENT '建设单位',

`construction\_agency` varchar(255) DEFAULT NULL COMMENT '代建单位',

`bid\_agency` varchar(255) DEFAULT NULL COMMENT '招标代理',

`design\_unit` varchar(255) DEFAULT NULL COMMENT '设计单位',

`survey\_unit` varchar(255) DEFAULT NULL COMMENT '勘察单位',

`control\_unit` varchar(255) DEFAULT NULL COMMENT '监理单位',

`property\_supervision\_unit` varchar(255) DEFAULT NULL COMMENT '财物监理单位',

`construction\_unit` varchar(255) DEFAULT NULL COMMENT '施工单位',

`testing\_unit` varchar(255) DEFAULT NULL COMMENT '材料检测单位',

`Parallel\_unit` varchar(255) DEFAULT NULL COMMENT '平行检测单位',

`total\_project\_cycle` varchar(255) DEFAULT NULL COMMENT '工程项目总周期',

`project\_description` varchar(255) DEFAULT NULL COMMENT '工程项目说明',

`project\_location` varchar(255) DEFAULT NULL COMMENT '工程位置',

`longitude` float DEFAULT NULL COMMENT '经度',

`latitude` float DEFAULT NULL COMMENT '纬度',

`total\_investment` decimal(12,4) DEFAULT NULL COMMENT '项目总投资',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '更新时间',

`tombstone` int unsigned DEFAULT '0' COMMENT '逻辑删除',

`phase` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT '项目书阶段' COMMENT '项目阶段',

`creator` varchar(255) DEFAULT NULL COMMENT '创建人',

`updateor` varchar(255) DEFAULT NULL COMMENT '修改人',

`total\_amount` decimal(10,2) DEFAULT NULL COMMENT '总金额',

`type` int NOT NULL DEFAULT '0' COMMENT '台账类型0-代建工程 1-总包工程',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=55 DEFAULT CHARSET=utf8mb3 COMMENT='代建工程主表';

CREATE TABLE `en\_project\_log` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT 'id',

`agent\_id` int DEFAULT NULL COMMENT '主表id',

`project\_name` varchar(255) DEFAULT NULL COMMENT '项目名称',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '更新时间',

`tombstone` int DEFAULT '0' COMMENT '逻辑删除',

`creator` varchar(255) DEFAULT NULL COMMENT '创建人',

`updateor` varchar(255) DEFAULT NULL COMMENT '更新人',

`file` varchar(255) DEFAULT NULL COMMENT '文件',

`content` varchar(255) DEFAULT NULL COMMENT '项目内容',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8mb3 COMMENT='代建项目日志';

CREATE TABLE `en\_operation\_log` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT 'id',

`operator` varchar(255) DEFAULT NULL COMMENT '操作人',

`operationTime` datetime DEFAULT NULL COMMENT '操作时间',

`operationModule` varchar(255) DEFAULT NULL COMMENT '操作模块',

`operationDescription` varchar(255) DEFAULT NULL COMMENT '操作描述',

`old` varchar(255) DEFAULT NULL COMMENT '旧的操作内容',

`new\_content` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '新的操作内容',

`updateTime` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8mb3 COMMENT='操作日志';

CREATE TABLE `en\_construction\_stage` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT 'id',

`project\_phase\_matters` varchar(255) DEFAULT NULL COMMENT '项目阶段及事项',

`proposed\_time` varchar(255) DEFAULT NULL COMMENT '拟定时间',

`responsible\_unit` varchar(255) DEFAULT NULL COMMENT '负责单位',

`cooperating\_unit` varchar(255) DEFAULT NULL COMMENT '配合单位',

`approval\_unit` varchar(255) DEFAULT NULL COMMENT '审核单位',

`keynote\_matters` varchar(255) DEFAULT NULL COMMENT '重点事项',

`state` int DEFAULT NULL COMMENT '状态（0未开始）',

`notes` varchar(255) DEFAULT NULL COMMENT '备注',

`agent\_id` int DEFAULT NULL COMMENT '主表id',

`Phases\_stare` int DEFAULT NULL COMMENT '阶段（0施工前，1施工中，2施工后）',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '更新时间',

`updateor` varchar(255) DEFAULT NULL COMMENT '更新人',

`creator` varchar(255) DEFAULT NULL COMMENT '创建人',

`tombstone` int DEFAULT '0' COMMENT '逻辑删除',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`stage\_name` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '阶段',

`project\_proposal\_picture` varchar(255) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '项目书图片id',

`other\_picture` varchar(255) DEFAULT NULL COMMENT '其他图片id',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=176 DEFAULT CHARSET=utf8mb3 COMMENT='代建阶段表';

CREATE TABLE `en\_construction\_fund` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '代建项目资金表id',

`agent\_id` int DEFAULT NULL COMMENT '主表id',

`year` varchar(255) DEFAULT NULL COMMENT '年',

`budget` decimal(10,2) DEFAULT NULL COMMENT '预算',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '更新时间',

`tombstone` int DEFAULT '0' COMMENT '逻辑删除',

`creator` varchar(255) DEFAULT NULL COMMENT '创建人',

`updateor` varchar(255) DEFAULT NULL COMMENT '更新人',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=utf8mb3 COMMENT='代建工程资金表';

CREATE TABLE `en\_engineering\_template\_stage\_matters` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`stage\_info\_id` int NOT NULL DEFAULT '0' COMMENT '阶段信息表id',

`matters\_name` varchar(255) NOT NULL COMMENT '阶段信息名字',

`sort` int NOT NULL DEFAULT '0' COMMENT '排序 数字越小顺序越高',

`delete\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识：0-未删、1-删除',

`creator` varchar(100) DEFAULT '' COMMENT '创建人',

`updator` varchar(255) DEFAULT '' COMMENT '修改人',

`create\_time` timestamp NULL DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=169 DEFAULT CHARSET=utf8mb3 COMMENT='工程模板阶段事项表';

CREATE TABLE `en\_engineering\_template\_stage\_info` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`main\_id` int NOT NULL DEFAULT '0' COMMENT '模板主表id',

`stage` int NOT NULL DEFAULT '0' COMMENT '阶段状态枚举0-施工前1-施工中2-施工后',

`stage\_name` varchar(255) NOT NULL COMMENT '阶段信息名字',

`delivery\_json` text COMMENT '交付文件json 格式是list',

`sort` int NOT NULL DEFAULT '0' COMMENT '排序 数字越小顺序越高',

`delete\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识：0-未删、1-删除',

`creator` varchar(100) DEFAULT '' COMMENT '创建人',

`updator` varchar(255) DEFAULT '' COMMENT '修改人',

`create\_time` timestamp NULL DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=146 DEFAULT CHARSET=utf8mb3 COMMENT='工程模板阶段信息表';

CREATE TABLE `en\_engineering\_template\_main` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`template\_name` varchar(255) NOT NULL COMMENT '工程名字',

`status` int NOT NULL DEFAULT '0' COMMENT '状态0-不启用1-启用',

`delete\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识：0-未删、1-删除',

`creator` varchar(100) DEFAULT '' COMMENT '创建人',

`updator` varchar(255) DEFAULT '' COMMENT '修改人',

`create\_time` timestamp NULL DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=38 DEFAULT CHARSET=utf8mb3 COMMENT='工程模板主表';

CREATE TABLE `en\_engineering\_contract\_ledger\_main` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`engineering\_id` int NOT NULL DEFAULT '0' COMMENT '工程关联id',

`signing\_time` datetime DEFAULT NULL COMMENT '签订日期',

`contract\_name` varchar(100) DEFAULT '' COMMENT '合同名称',

`party\_a` varchar(100) DEFAULT '' COMMENT '甲方名称',

`party\_b` varchar(100) DEFAULT '' COMMENT '乙方名称',

`contract\_code` varchar(255) DEFAULT NULL COMMENT '合同编号',

`contract\_amount` decimal(64,4) NOT NULL COMMENT '合同金额',

`review\_amount` decimal(64,4) NOT NULL COMMENT '审价金额',

`auditing\_amount` decimal(64,4) NOT NULL COMMENT '审计金额',

`construction\_time` varchar(255) DEFAULT NULL COMMENT '施工期',

`unpaid\_amount` decimal(64,4) DEFAULT NULL COMMENT '未支付金额',

`review\_time` datetime DEFAULT NULL COMMENT '审价日期',

`taxes` decimal(12,2) DEFAULT NULL COMMENT '税金',

`accumulate\_payment\_amount` decimal(64,4) DEFAULT NULL COMMENT '合同累计支付金额',

`accumulate\_payment\_proportion` decimal(64,4) DEFAULT NULL COMMENT '合同累计支付比例',

`accumulate\_payment\_term` text COMMENT '合同累计支付条款',

`related\_contract` varchar(255) DEFAULT NULL COMMENT '关联合同',

`notes` text COMMENT '备注',

`file\_ids` text CHARACTER SET utf8 COLLATE utf8\_general\_ci COMMENT '上传比价情况文件 多个用,逗号拼接',

`type` int NOT NULL DEFAULT '0' COMMENT '台账类型0-代建工程 1-总包工程',

`party\_type` int NOT NULL DEFAULT '0' COMMENT '总包合同台账类型1- 甲方 2-乙方',

`delete\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识：0-未删、1-删除',

`creator` varchar(100) DEFAULT '' COMMENT '创建人',

`updator` varchar(255) DEFAULT '' COMMENT '修改人',

`create\_time` timestamp NULL DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=110 DEFAULT CHARSET=utf8mb3 COMMENT='合同台账信息主表';

CREATE TABLE `en\_engineering\_contract\_ledger\_amount\_details` (

`id` int NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`ledger\_id` int NOT NULL DEFAULT '0' COMMENT '台账关联id',

`Invoicing\_time` datetime DEFAULT NULL COMMENT '开票日期',

`payment\_amount` decimal(64,4) NOT NULL COMMENT '本期支付金额',

`file\_ids` varchar(100) DEFAULT '' COMMENT '付款资料 多个用,逗号拼接',

`revenue\_status` varchar(255) DEFAULT NULL COMMENT '收入状态',

`delete\_flag` tinyint(1) DEFAULT '0' COMMENT '删除标识：0-未删、1-删除',

`creator` varchar(100) DEFAULT '' COMMENT '创建人',

`updator` varchar(255) DEFAULT '' COMMENT '修改人',

`create\_time` timestamp NULL DEFAULT NULL COMMENT '创建时间',

`update\_time` timestamp NULL DEFAULT NULL ON UPDATE CURRENT\_TIMESTAMP COMMENT '修改时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=108 DEFAULT CHARSET=utf8mb3 COMMENT='合同台账信息本期支付信息表';