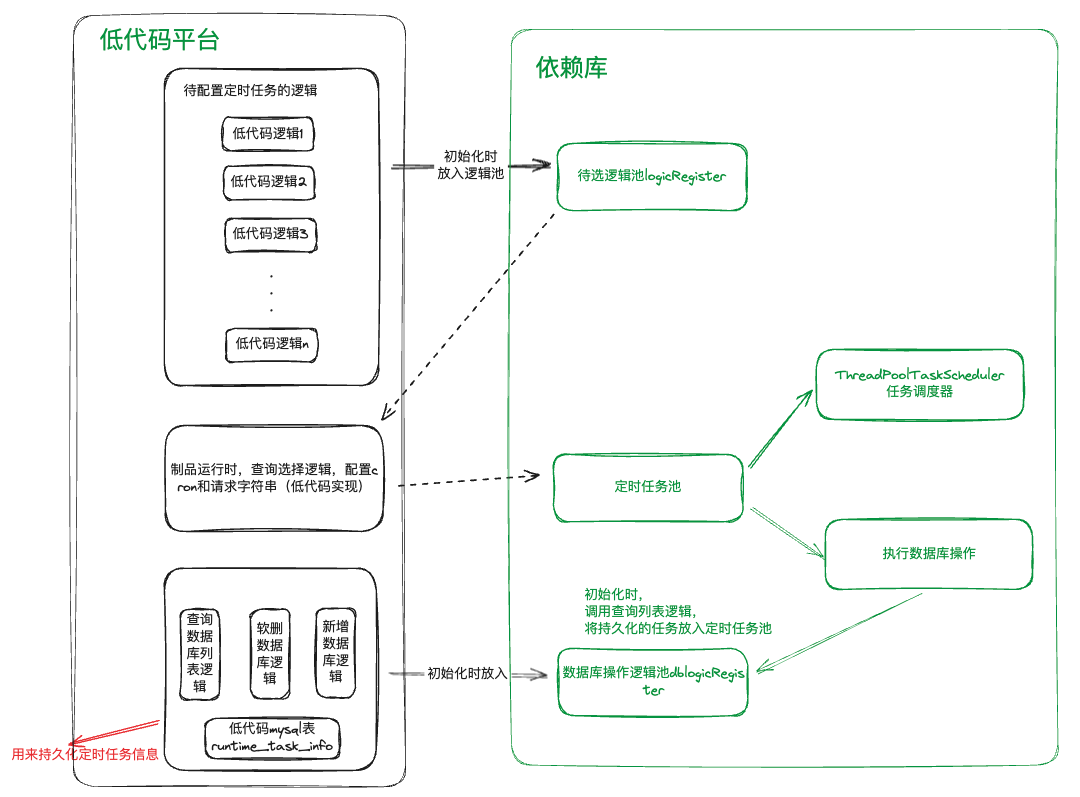
**一、背景说明**

当前平台的定时任务能力基于quartz框架。在可视化界面中，可以为某个逻辑设置定时任务的表达式。当制品启动时，根据表达式生成相应的定时任务代码。每个任务对应的逻辑会生成一个handler/job类，该类中包含一个execute方法用于执行任务内容（即逻辑内容）。然而，由于它是基于代码的，仅允许用户在编译前管理定时任务。因此无法在运行时满足客户的需求，即无法动态增加或删除任务。

**二、设计思路**



1. 手动讲需要定时的logic放入注册到逻辑注册器（依赖库logicRegister）。调用依赖库方法，将logic传入。

2. 查询已注册的逻辑列表。

3. 对选择的逻辑，配置cron，创建任务，数据放入ThreadPoolTaskScheduler、任务注册器和数据库。

4. 不支持查看任务执行记录，需要查看时，需要在任务逻辑内部做操作。

5. 删除任务时，从ThreadPoolTaskScheduler中cancel，从任务注册器和数据库中删除。

6. 查询已创建的任务列表。

7. 制品重启时，会先执行1对应的代码，再从数据库中获取配置的任务，放入ThreadPoolTaskScheduler和任务注册器。

**三、接口说明（非最终版本）**

1. **初始化注册任务（持久化）registerInit**

本接口需要在初始化时执行。它用于注册3个定时任务持久化逻辑，逻辑须在IDE实现操作数据库。并且执行logicFunctionList获取当前数据库中所有定时任务，加载到内存中，触发定时任务的启动。IDE中相关操作见使用说明。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicFunctionList | 查询定时任务列表逻辑 | Function<Srting,String> | IDE服务端逻辑 |
| logicFunctionAdd | 新增定时任务逻辑 | Function<Srting,String> | IDE服务端逻辑 |
| logicFunctionDelete | 根据taskId删除定时任务逻辑 | Function<Srting,String> | IDE服务端逻辑 |

出参：true/false

1. **注册逻辑addLogic**

本接口用于把IDE中需要供用户配置定时任务的逻辑，注册到逻辑池中。注册到逻辑池的逻辑，才可以被应用使用者配置任务执行规则。本接口需要在初始化时将所有逻辑注册成功。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicFunction | 待配置定时的逻辑 | Function<Srting,String> | IDE服务端逻辑，不可空 |
| logicName | 逻辑名称 | String | 自定义，不可空 |

出参：true/false

1. **查询所有逻辑(右模糊匹配)listLogic**

本接口用于查询逻辑池中所有逻辑。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicName | 逻辑名称 | String | 可空。为空时查询全部。 |

出参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
|  | 逻辑名称列表 | List<String> |  |

1. **立即执行逻辑runLogic**

本接口用于执行逻辑池中某个逻辑，可传入入参。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicName | 逻辑名称 | String | 可空。为空时查询全部。 |
| request | 入参字符串 | String | 要传入逻辑的字符串 |

出参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
|  | 执行结果 | String |  |

1. **创建定时任务createTask**

通过本接口将逻辑和cron、入参字符串传入，即可创建定时任务。若接口1传入了持久化逻辑，定时任务会持久化到数据库。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicName | 逻辑名称 | String | 可空。为空时查询全部。 |
| cron | cron表达式/任务启动时间YYYYMMDDhhmmss | String | 定时循环启动，或延迟启动。 |
| request | 入参字符串 | String | 要传入逻辑的字符串 |

出参：true/false

1. **查询已已执行的任务列表(右模糊匹配)listTask**

查询内存中已经启动的任务。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicName | 逻辑名称 | String | 可空。为空时查询全部。 |

出参：List<TaskStructure>

TaskStructure结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| logicName | 逻辑名称 | String | 可空。为空时查询全部。 |
| cron | cron表达式/任务启动时间YYYYMMDDhhmmss | String | 定时循环启动，或延迟启动。 |
| request | 入参字符串 | String | 要传入逻辑的字符串 |
| taskId | 任务id | String |  |

1. **删除任务deleteTask**

通过taskId删除定时任务。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| taskId | 任务id | String |  |

出参：true/false

1. **暂停任务pauseTask**

通过taskId删除定时任务。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| taskId | 任务id | String |  |

出参：true/false

1. **启动任务startTask**

通过taskId删除定时任务。

入参：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 描述 | 类型 | 说明 |
| taskId | 任务id | String |  |

出参：true/false

**四、使用说明**

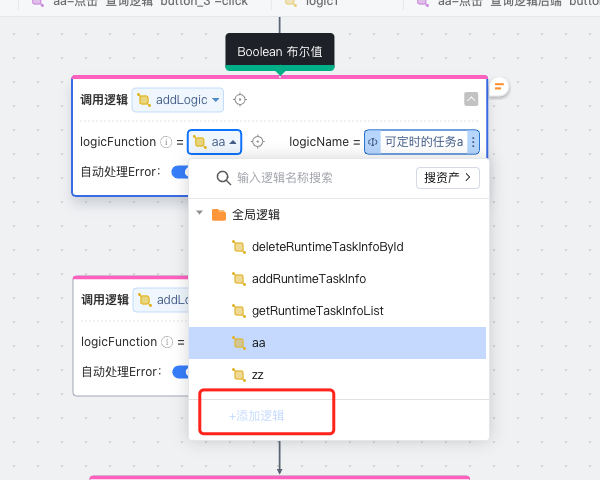
所有逻辑建议在服务端使用，因为业务异常时，服务端调用会返回正确提示信息，前端调用会500错误。

**步骤一：初始化逻辑池**

调用addLogic把待配置定时信息的逻辑传入依赖库。**须在初始化时调用。**

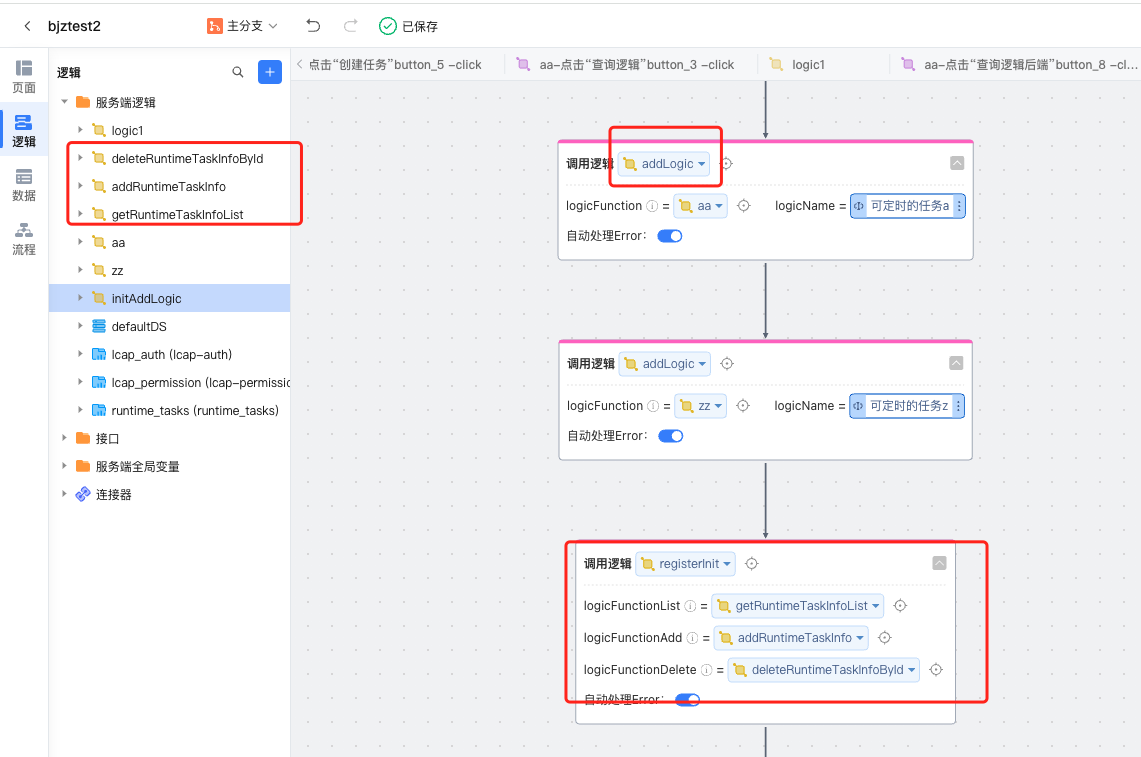


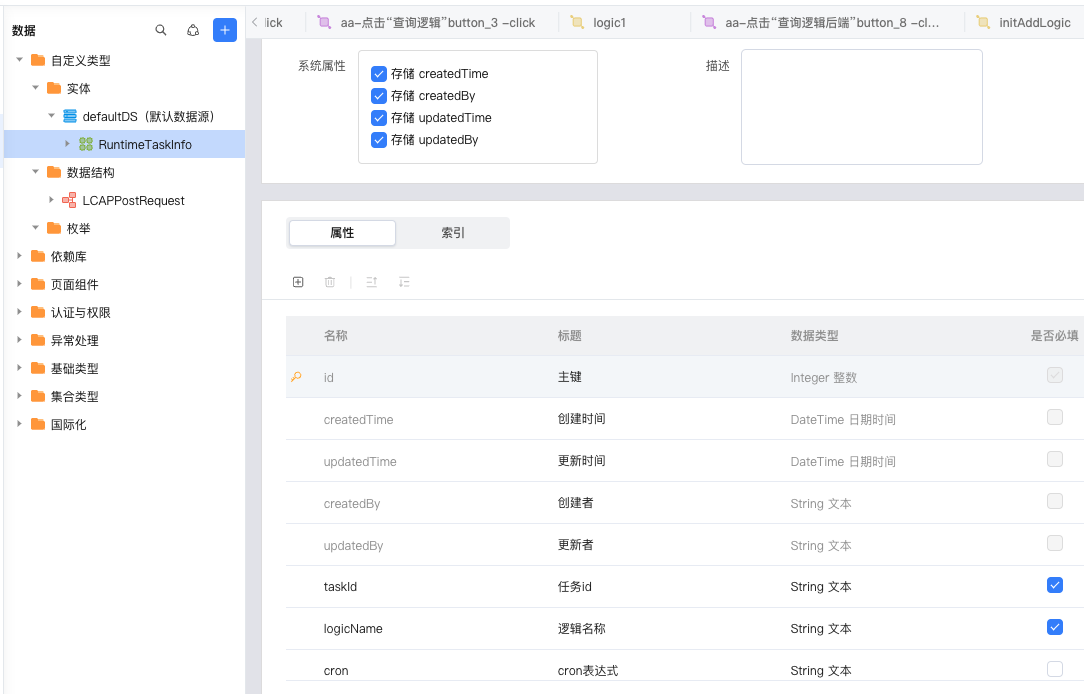
aa/zz这类要传入依赖库的逻辑，不能直接在服务端逻辑菜单中新建。需要在调用addLogic页面中添加逻辑，如下图。



**步骤二：初始化任务持久化**

调用registerInit传入操作表RuntimeTaskInfo的三个逻辑（局部模板支持）





**步骤三：开发定时任务管理页面，供制品使用者在运行时操作任务**

3.1 listLogic查询所有逻辑

3.2 runLogic测试逻辑可行性

3.3 createTask选择一个逻辑，为其配置cron和请求入参

3.4 listTask查看所有的已配置任务

3.5 deleteTask删除已配置任务

3.6 pauseTask和startTask暂停启动任务