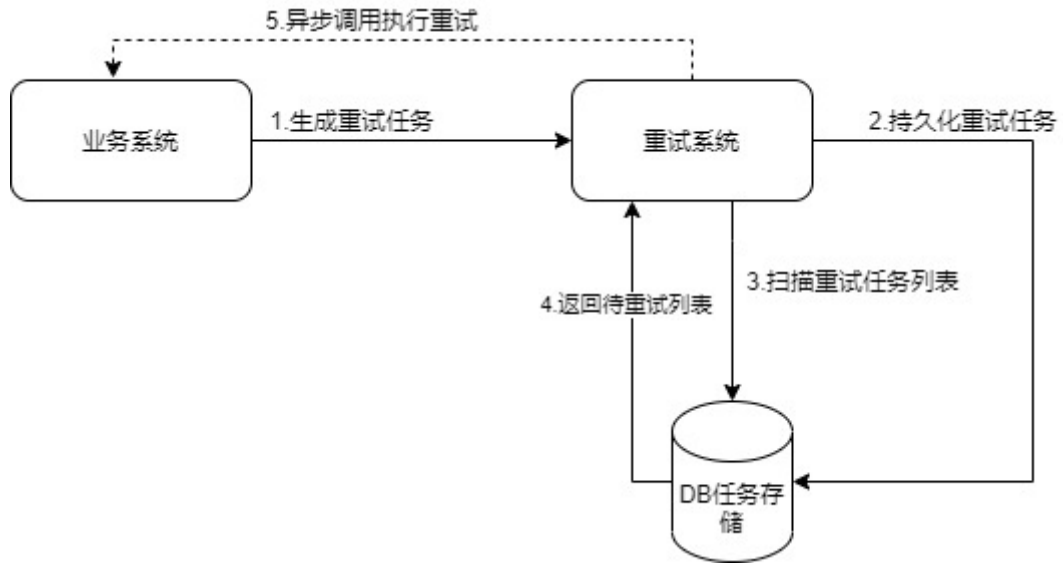
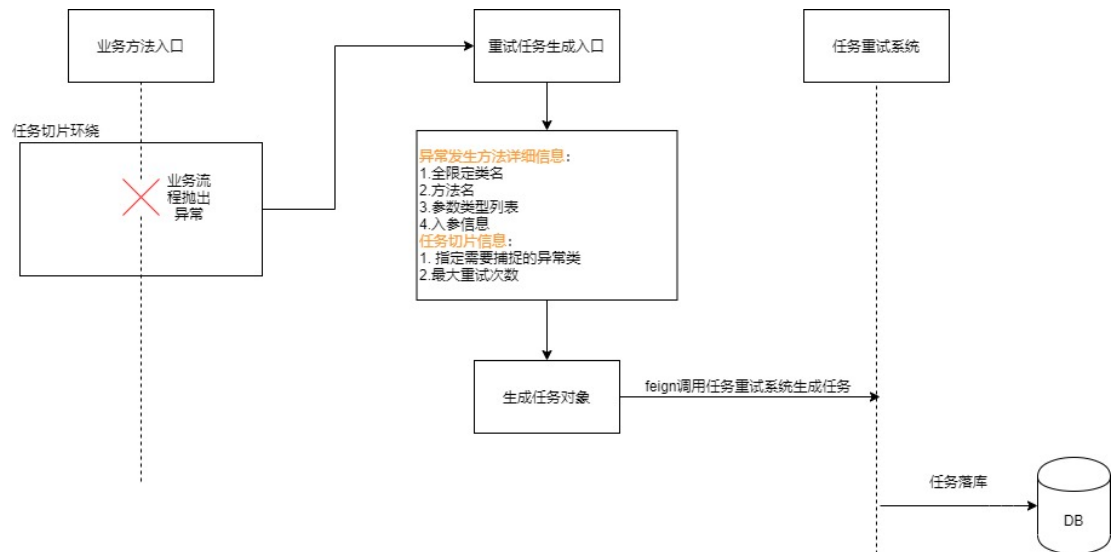


## 1. 系统交互

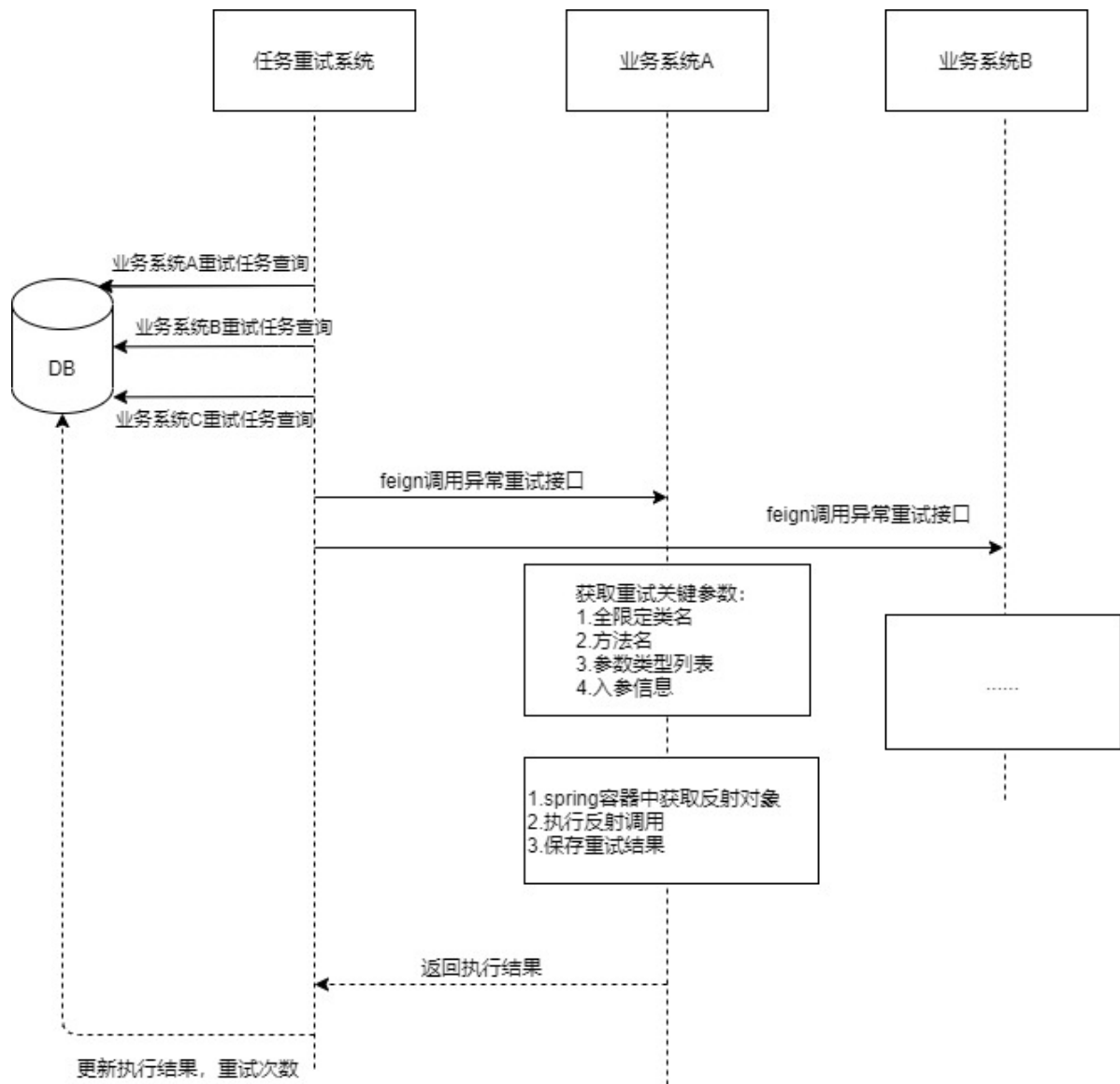


## 2. 时序图

### • 生成任务



### • 重试任务



### 3. 重试系统表结构

```

CREATE TABLE `task` (
  `id` bigint(30) NOT NULL AUTO_INCREMENT,
  `application_name` varchar(50) DEFAULT NULL COMMENT '业务系统名称',
  `full_class_name` varchar(200) NOT NULL COMMENT '方法所在的全限定类名',
  `method_name` varchar(100) NOT NULL COMMENT '重试的方法名',
  `parameter_types` text COMMENT '参数类型',
  `parameters` text COMMENT '具体参数',
  `status` int(11) NOT NULL DEFAULT '0' COMMENT '任务状态,0:初始化 1:执行中
2:执行成功 3:执行失败 4:无效任务',
  `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `last_modify_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP COMMENT '修改时间',
  `execute_times` int(11) NOT NULL DEFAULT '0' COMMENT '执行次数',
  `version` int(11) NOT NULL DEFAULT '0' COMMENT '版本号',
  `remark` text COMMENT '方法执行错误信息',
  `max_retry_time` int(11) NOT NULL COMMENT '最大重试次数',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8
  
```

#### 4. QA

- 为什么需要写重试组件？

分布式系统中远程调用存在网络问题、超时问题等，进行任务判断可以进行重试时便可由系统自动完成，无需人为介入刷数据

- 为什么不使用spring自带retry组件？

springretry组件是同步，若配置重试时长过长会增加方法执行耗时，增大OOM的几率，并且会引起上游调用超时。

确实存在下游网关停机，内存同步重试几乎无用，需要有异步重试的组件。

实际开发中应该内存同步重试+异步重试组合使用

- 为什么不使用xxl-job api动态生成任务？

xxl-job的重试时为了解决一类问题，人为知道了需要重试哪一个具体的事情，任务重试组件不关心具体类型的事情，只要能被spring管理到就能进行重试。