# 批量导入异步任务框架

**需求说明：**提供批量导入异步任务框架，支持各类业务数据的批量导入(例如工单、工单类型等)。具体功能点包括：模板下载、模板标题批注、excel数据导入、导入进度查询、导入结果下载等，并且数据导入规则支持扩展。

**实现思路：**浏览器客户端上传文件到Web Portal，Web Portal上的FileUploadService调用OBS的客户端SDK将文件上传到OBS，并返回Object Key给客户端。客户端根据Object Key再调用Case APP的批量导入接口，执行批量导入。

Case APP上，根据Object Key下载文件，然后批量导入。

Case APP上将每个导入任务封装为一个BatchImportRunnable对象，由线程池调度执行。任务将包含Object Key、Object Name等属性，通过工厂模式根据Object Name实例化对应的RowData、FieldConfiguration等。提供一个AbstractBatchImportService抽象类，包含batchImport方法和doBatchImport方法，batchImport方法里执行文件下载、excel解析为行数据集合、doBatchImport以及最后的生成导入结果文件并上传文件服务器等操作，doBatchImport执行具体的导入逻辑，它是abstract protected的，由子类实现，这里运用了模板方法模式。

为了防止并发导入时插入重复数据，在doBatchImport的前面会先进行分布式锁的申请，分布式锁使用了单机redis的setNX操作来实现。

批量导入最终都会调用Bingo的批量插入接口实现数据的批量入库，如果导入很多数据，势必造成Bingo入库时间较长，Bingo的Nginx会返回504网关超时。为了解决这个问题，框架又提供了一个BatchInsertDelegate工具类，使用时直接传入全量的待插入的数据，BatchInsertDelegate将使用Fork-Join框架进行子任务拆分，拆成<=1000的数据量，再发送请求给Bingo，具体的拆分方式我使用的是二分法递归拆分。

在开发过程中，由于文件存储方案不确定，一会要求OBS存储，一会又要求LSS存储，框架又抽象出一个StorageService接口，包含upload接口(向文件服务器上传文件)和download接口(从文件服务器下载文件)，OBS存储的话就提供OBSStorageService Bean实现之，里面调用OBS的客户端SDK实现存储和下载，LSS的话就提供LSSStorageService Bean实现该接口，里面调用LSS的接口实现上传下载，客户端直接依赖的是StorageService，根据类型自动装配存储服务bean，然后具体要用的存储服务Bean上加上@Primary标记优先。

遇到一个坑点是，为了在线程抛出未受检异常或者Error时能够更新后台任务的状态为失败，我在线程池的ThreadFactory的newThread方法里把线程名和后台任务ID进行了关联—线程名都叫”servicecloud-bulkbiz-${backgroundTaskId}”这种格式，然后设置线程的UncaughtExceptionHandler从线程名上取后台任务ID然后更新表，但是在做异常场景测试的时候发现后台尽管已经抛了异常，却还是无法更新对应的任务状态为失败。后来发现是由于线程池核心池里的线程复用导致的，新任务的创建没有调用ThreadFactory的newThread创建新线程，而是复用的旧线程，因此线程名上的ID和任务的真实ID不匹配，导致更新错误。解决办法也很简单，就是把线程名与backgroundTaskId的关联的操作放到Runnable的run里去做就行，不能放在线程工厂的newThread里。

**技术要点：**多线程、线程池、ThreadLocal、工厂模式、模板方法模式、分布式锁、HashMap进行等价类划分、ConcurrentHashMap(单例的注册中心Bean里通过ConcurrentHashMap保存各个单例的注册项Bean)、针对接口编程、Fork-Join框架。

**可提升点：**

1. Bingo不支持事务，而业务场景上存在先批量入库A数据再批量入库B数据的情况，因此要实现事务补偿。