**关于自由报表导入技术方案概述**

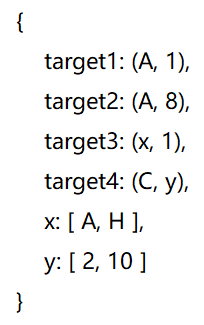
1. **概述**

我们在分析了 1104 报表模板以及 MPA 报表模板之后，发现各种模板的原始表转换成我们需要的目标表时，都符合这样一种形式：

**Target(x`, y`) <= Source(x, y)**

即：目标表某一列数据取自于原始表种指定单元格或某一范围的单元格。

因此，针对每一个模板都需要有一份类似如下所示的配置文件：、



其中，targetX表示目标表中相应的字段，x表示行的范围，y表示列的范围，(\*, \*)表示原始表中某个点，[\*, \*]表示原始表中某个范围。

综上，就维护了从原始表到目标表的转换关系，因此在 Java 代码中针对转换关系做统一处理，就能完成兼容各种模板的数据导入。

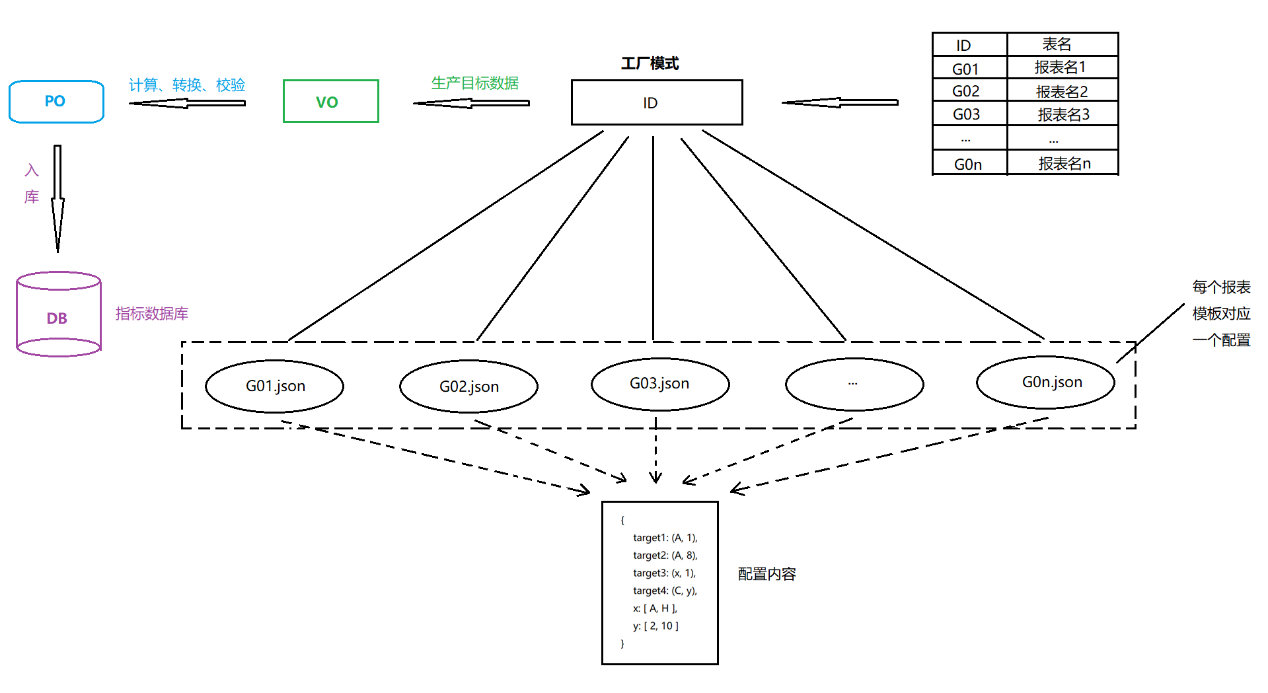
1. **后台设计**

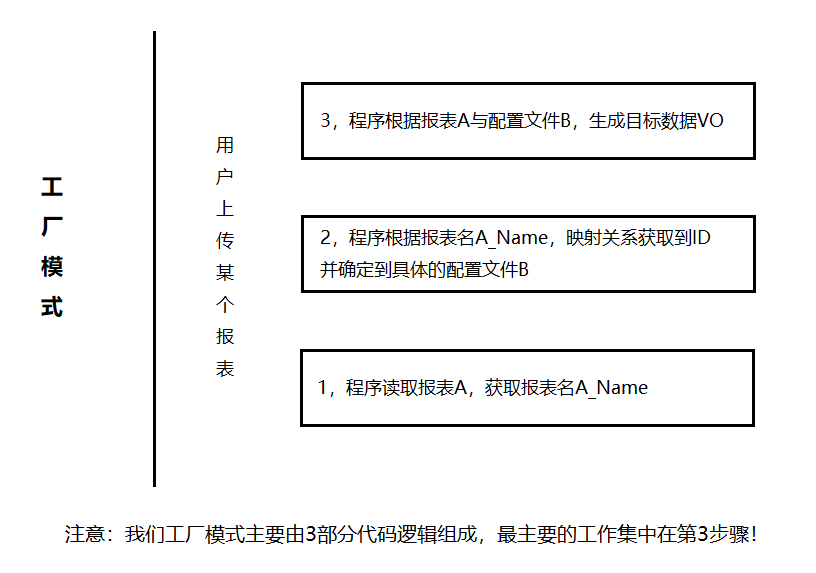
后台代码开发时，采用工厂模式，针对用户上传的报表 DataReport 识别出对应的 ID 并传入工厂，工厂中根据 ID在配置目录中匹配到对应的目标配置文件并解析形成配置对象ConfigObj。

工厂模式的代码逻辑中提供DataReport与ConfigObj相结合形成最终VO的方法，这样就完成不同的原始报表到实体VO的解析。

针对VO，依据相关的业务逻辑对VO的数据进行清洗（计算、转换、校验），最终形成PO并入库。

综上，如下图所示：





1. **附加事项**

根据以上分析，有以下两点需要后期具体需求具体细化：

* VO与PO的具体结构，需要符合所有的模板表以及数据库指标数据表；
* 配置文件 \*.json 的具体配置项目；

1. **例子**

略