

# WorkflowServiceImpl.java

## 数据流SQL转换分析

`\supersonic\headless\server\src\main\java\com\tencent\supersonic\headless\server\web\service\impl\WorkflowServiceImpl.java`

`WorkflowServiceImpl` 的主要职责是协调查询处理的各个步骤，包括映射、解析、校正和结果处理。通过这些步骤，查询上下文逐步完善，最终生成执行结果。

## 类的主要方法

`startWorkflow(QueryContext queryCtx, ChatContext chatCtx, ParseResp parseResult)`

这个方法是工作流的入口。它通过一个循环，依次执行工作流中的不同阶段，直到整个工作流结束（`WorkflowState.FINISHED`）。

1. **初始状态**：方法一开始将 `QueryContext` 的状态设置为 `WorkflowState.MAPPING`，即开始映射阶段。
2. **状态机驱动**：根据当前的状态，依次调用 `performMapping`、`performParsing`、`performCorrecting` 和 `performProcessing` 方法。
3. **状态切换**：每个阶段执行完后，会根据状态机的设计将状态切换到下一个阶段，直到所有步骤完成并将状态设置为 `WorkflowState.FINISHED`。

`performMapping(QueryContext queryCtx)`

映射阶段的主要任务是将用户的查询与语义模型中的元素进行匹配。这一步主要通过 `schemaMappers` 完成：

- **映射检查**：首先检查 `QueryContext` 中是否已有映射信息。如果没有，则依次调用所有的 `SchemaMapper` 实例来执行映射操作。

`performParsing(QueryContext queryCtx, ChatContext chatCtx)`

解析阶段负责将映射的结果转换为 SQL 查询语句：

- **调用解析器**：对所有的 `SemanticParser` 实例进行调用，执行解析操作。
- **日志记录**：在每个解析器完成解析后，记录当前上下文的状态，便于调试和分析。

`performCorrecting(QueryContext queryCtx)`

校正阶段是对解析结果进行调整和优化：

- **校正器调用**：遍历所有的候选查询，调用 `SemanticCorrector` 实例进行校正。
- **状态控制**：在校正过程中，如果状态从 `CORRECTING` 发生变化，则跳出校正循环，继续执行下一个步骤。

`performProcessing(QueryContext queryCtx, ChatContext chatCtx, ParseResp parseResult)`

处理阶段是对解析和校正后的查询进行处理并生成最终的响应结果：

- **结果处理器调用**：遍历所有的 `ResultProcessor` 实例，将查询上下文和校正后的信息进行最终处理，生成 `ParseResp` 对象。

## 依赖的组件

- `SchemaMapper`：用于将用户的查询映射到语义模型中的元素，比如字段、表、度量等。
- `SemanticParser`：负责将映射结果解析为 SQL 查询。
- `SemanticCorrector`：对生成的 SQL 查询进行校正和优化。
- `ResultProcessor`：负责处理最终的 SQL 查询，生成查询结果。

## 状态管理

`WorkflowServiceImpl` 通过 `WorkflowState` 枚举来管理查询处理的不同阶段。状态在每个步骤结束后切换，从而确保各个阶段按顺序执行。

- `MAPPING`：初始状态，表示正在进行映射。
- `PARSING`：映射完成后进入解析阶段。
- `CORRECTING`：解析完成后进入校正阶段。
- `PROCESSING`：校正完成后进入处理阶段。
- `FINISHED`：工作流结束。

## 总结

`WorkflowServiceImpl` 是整个查询处理流程的协调者，它确保了用户查询从初始输入到最终生成查询结果的整个流程得以顺利进行。通过模块化的设计，它将各个阶段的处理逻辑分离到不同的组件中，保证了代码的可扩展性和维护性。