

SchemaServiceImpl

`SchemaServiceImpl` 类主要负责处理与语义架构（Semantic Schema）相关的服务逻辑。这类服务包括获取数据集的模式、构建语义架构响应、缓存管理等功能。以下是对该类及其功能的详细分析：

类的核心功能

`SchemaServiceImpl` 提供了以下核心功能：

1. 获取数据集模式 (`fetchDataSetSchema`) :
 - 该方法根据给定的 `DataSetFilterReq` 过滤条件来获取数据集的模式信息。如果启用了缓存，则首先从缓存中获取数据，否则通过调用 `buildDataSetSchema` 方法来构建模式信息，并将结果存入缓存中。
2. 获取单个数据集模式 (`fetchDataSetSchema(Long dataSetId)`) :
 - 这个方法用于根据数据集ID获取特定的数据集模式。
3. 构建数据集模式 (`buildDataSetSchema`) :
 - 该方法根据 `DataSetFilterReq` 构建完整的数据集模式。包括获取数据集的基本信息、维度、度量、模型等信息，并整合它们构建最终的响应对象。
4. 获取语义架构 (`getSemanticSchema`) :
 - 这个方法返回所有数据集的语义架构的完整视图。
5. 构建语义架构响应 (`buildSemanticSchema`) :
 - 该方法用于根据给定的过滤条件构建语义架构响应对象 `SemanticSchemaResp`，其中包含模型、维度、度量和相关关系等信息。
6. 获取统计信息 (`getStatInfo`) :
 - 该方法用于获取使用统计信息，并支持缓存。
7. 获取域数据集树 (`getDomainDataSetTree`) :
 - 该方法获取域与数据集的树状结构，便于在前端展示。
8. 从 YAML 模板构建架构 (`getSchemaYamlTpl`) :
 - 这个方法用来将语义架构响应对象转换为 YAML 模板，适合用于配置文件的生成和管理。

与其他组件的关联

1. 与缓存的关联：

- 该类使用 Guava 的 `Cache` 对象来管理数据集模式和语义架构的缓存，从而提高查询的效率，减少对数据库或其他数据源的直接访问频率。

2. 与其他服务的依赖：

- `SchemaServiceImpl` 依赖于多个其他服务，如 `ModelService`、`DimensionService`、`MetricService` 等。这些服务负责提供与模型、维度、度量、数据库等相关的数据操作接口。

3. 与 `QueryTypeParser` 和 `LLMSqlParser` 的关系：

- 尽管 `SchemaServiceImpl` 主要聚焦于语义架构和数据集模式的管理，但它提供的语义架构（例如通过 `getSemanticSchema()` 方法）可以被其他组件（如 `QueryTypeParser` 和 `LLMSqlParser`）使用，以便在查询解析和类型识别过程中获取必要的语义信息。

工作流程示例

• 数据集模式获取流程：

1. 用户请求获取特定数据集的模式信息。
2. `SchemaServiceImpl` 调用 `fetchDataSetSchema` 方法，首先检查缓存，如果缓存中不存在该数据集的模式信息，则调用 `buildDataSetSchema` 方法从基础数据源构建该模式信息。
3. 构建完成后，将结果存入缓存并返回给调用方。

• 语义架构构建流程：

1. 系统需要生成某个数据集或模型的语义架构。
2. `SchemaServiceImpl` 调用 `buildSemanticSchema` 方法，从相关服务获取必要的信息（例如模型、维度、度量等），并将其整合为 `SemanticSchemaResp` 对象。
3. 生成的语义架构对象可以被缓存，也可以直接用于查询解析或其他逻辑中。