vistora

Crawler to Extract MySQL Database Schema and Build Models

Objective:

Create a Java-based crawler that connects to a MySQL database, crawls through its tables, columns, relationships, and other database metadata, and automatically generates model representations of the schema. The crawler should produce a programmatic model or object-oriented representation of the database schema (e.g., classes, relationships, and attributes).

Requirements:

1. **MySQL Database Crawler (Java):**
    - The crawler should connect to a MySQL database using JDBC.
    - The crawler should retrieve and extract metadata about the database schema, including:
        - **Tables**: List of tables in the database.
        - **Columns**: Column names, data types, lengths, nullable properties, and other constraints.
        - **Primary Keys**: Extract primary key information for each table.
        - **Foreign Keys**: Identify relationships between tables through foreign keys.
        - **Indexes**: Capture any indexes defined on the tables.
2. **Model Generation:**
    - Based on the extracted schema, the crawler should generate Java model classes dynamically (i.e., at runtime).
    - The model classes should represent tables as objects, with:
        - **Fields**: Representing columns with appropriate types (e.g., String, int, Date).
        - **Relationships**: Representing foreign key relationships between tables using appropriate object references (e.g., User class referencing Address class via a foreign key).
3. **Database Metadata:**
    - Ensure that the crawler can handle different types of databases with minimal configuration changes.
4. **REST APIs:**
    - **REST APIs: Develop REST APIs for Crawler to get Database metadata and model**
    - **Handle Complex Relationships**: If there are complex relationships like many-to-many (join tables), create models that represent this appropriately.
    - **Configuration File**: Allow users to configure the database connection and crawler behavior through a JSON configuration file.

Deliverables:

- REST API for the database crawler, its metadata and model generation logic.
- A sample output of the REST APIs based on an example database schema.
- A brief documentation explaining:
    - The crawler's workflow (how it connects to the database, extracts metadata, and generates models).