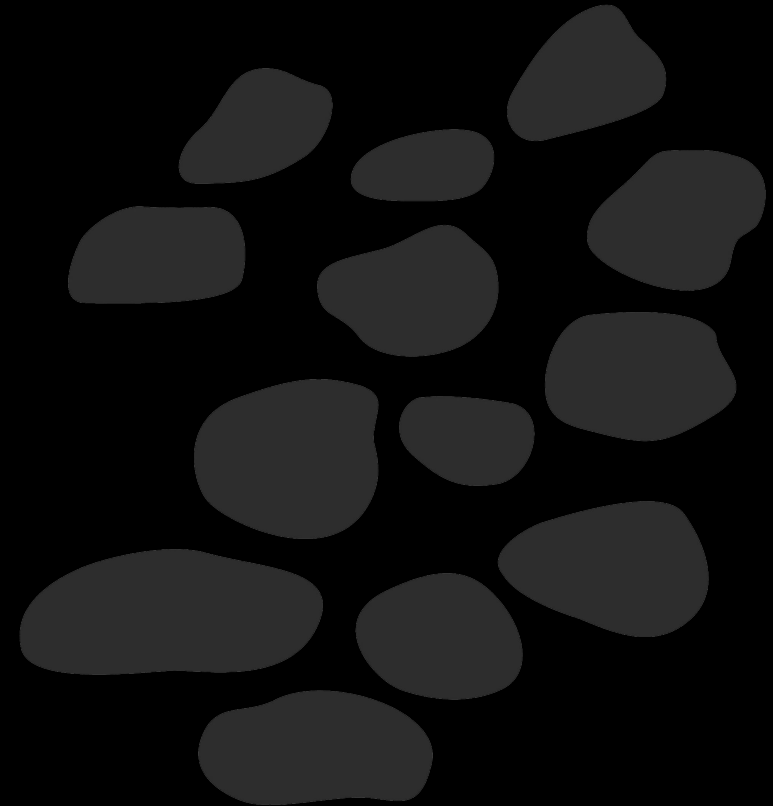


*gepard*ec

Camel - SQL



Why?

// The SQL component allows you to work with databases using JDBC queries.

// The difference between this component and JDBC component is that in case of SQL the query is a property of the endpoint and it uses message payload as parameters passed to the query.

Configure SQL Component

// Include maven dependencies

```
<dependency>
  <groupId>org.apache.camel.quarkus</groupId>
  <artifactId>camel-quarkus-sql</artifactId>
</dependency>
```

// Configure Datasources

```
quarkus.datasource.my_db.db-kind=postgresql
quarkus.datasource.my_db.username=camel
quarkus.datasource.my_db.password=mypassword
quarkus.datasource.my_db.jdbc.url=jdbc:postgresql://localhost:5432/camel
quarkus.datasource.my_db.jdbc.max-size=11
```

```
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-jdbc-postgresql</artifactId>
</dependency>
```

// Implement Component (optional)

Default is 'sql'

```
@Named("mypgsql")
public SqlComponent mypgsql(@Named("my_db") DataSource dataSource)
{
    SqlComponent mydb = new SqlComponent();
    mydb.setDataSource(dataSource);
    return mydb;
}
```

Configure Endpoint

```
sql://insert into my_table (col1, col2)
  values (:#par1, :#par2 )
```

Can be a key when
body is a Map

Can be a simple
expression

```
@Named
public IdGenerator uuidGen() {
    return new IdGenerator();
}
```

```
public class IdGenerator {
    public UUID nextId() {
        return UUID.randomUUID();
    }
}
```

```
.to("sql://insert into order_to_producer (id, partner_id, item_code, item_count) "
    + "values (CAST(:#{bean:uuidGen.nextId} AS uuid), :#{body.partnerId}, :#{body.code}, :#{body.amount})")
```

```
sql:classpath:sql/myquery.sql[?options]
```

Put query into file on
classpath

Exercise

best_12_1_sql_component

best_12_2_sql_insert