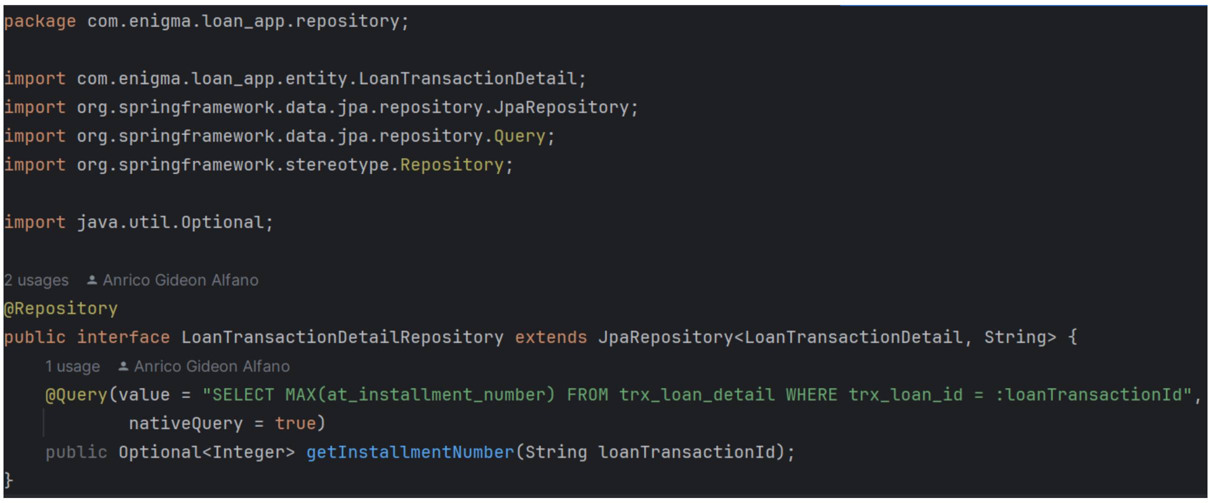
Native SQL Query and Java Stream Usage Documentation

1. Native SQL Query



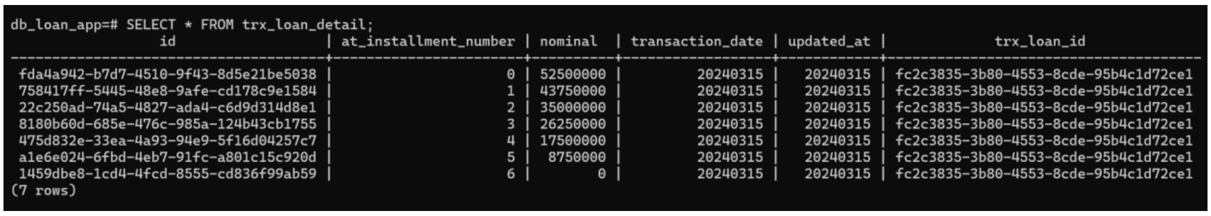
```java
package com.enigma.loan_app.repository;

import com.enigma.loan_app.entity.LoanTransactionDetail;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.Optional;

2 usages   Anrico Gideon Alfano
@Repository
public interface LoanTransactionDetailRepository extends JpaRepository<LoanTransactionDetail, String> {
    1 usage   Anrico Gideon Alfano
    @Query(value = "SELECT MAX(at_installment_number) FROM trx_loan_detail WHERE trx_loan_id = :loanTransactionId",
            nativeQuery = true)
    public Optional<Integer> getInstallmentNumber(String loanTransactionId);
}
```

This specific method (query) is easier to achieve with SQL Query so native query is used here. This method is used to generate number for instalment (at_installment_number field):

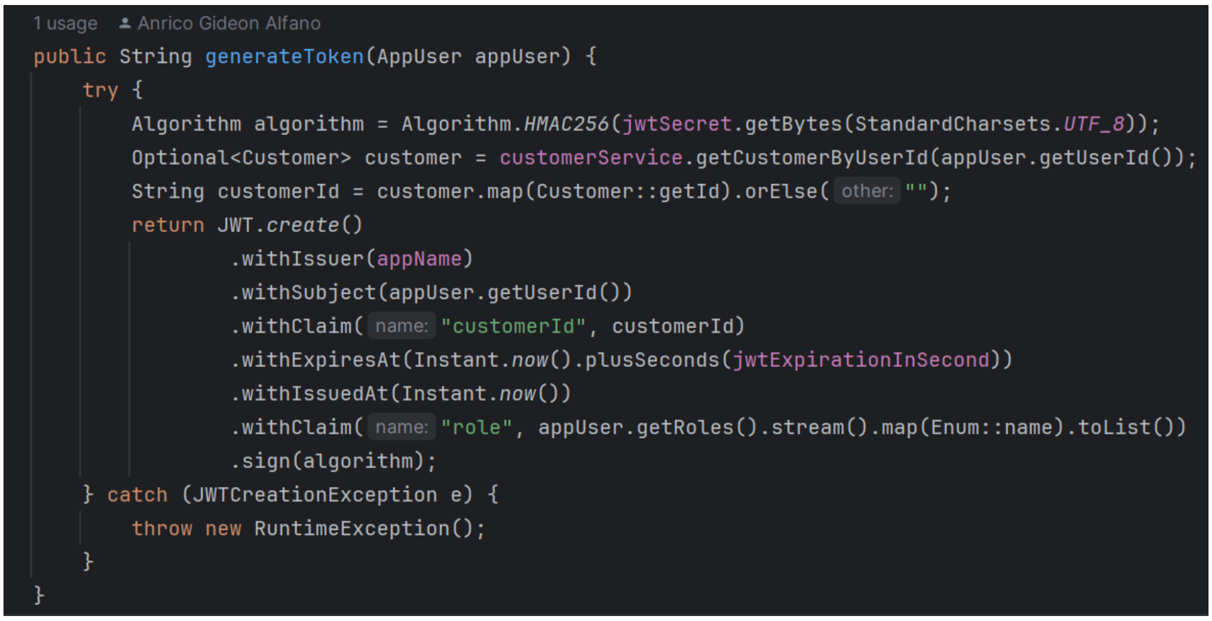


```
db_loan_app=# SELECT * FROM trx_loan_detail;
                  id                  | at_installment_number | nominal  | transaction_date | updated_at |             trx_loan_id
--------------------------------------+-----------------------+----------+------------------+------------+--------------------------------------
 fda4a942-b7d7-4510-9f43-8d5e21be5038 |                     0 | 52500000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 758417ff-5445-48e8-9afe-cd178c9e1584 |                     1 | 43750000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 22c250ad-74a5-4827-ada4-c6d9d314d8e1 |                     2 | 35000000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 8180b60d-685e-476c-985a-124b43cb1755 |                     3 | 26250000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 475d832e-33ea-4a93-94e9-5f16d04257c7 |                     4 | 17500000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 a1e6e024-6fbd-4eb7-91fc-a801c15c920d |                     5 |  8750000 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
 1459dbe8-1cd4-4fcd-8555-cd836f99ab59 |                     6 |        0 |         20240315 |   20240315 | fc2c3835-3b80-4553-8cde-95b4c1d72ce1
(7 rows)
```

2. Java Stream

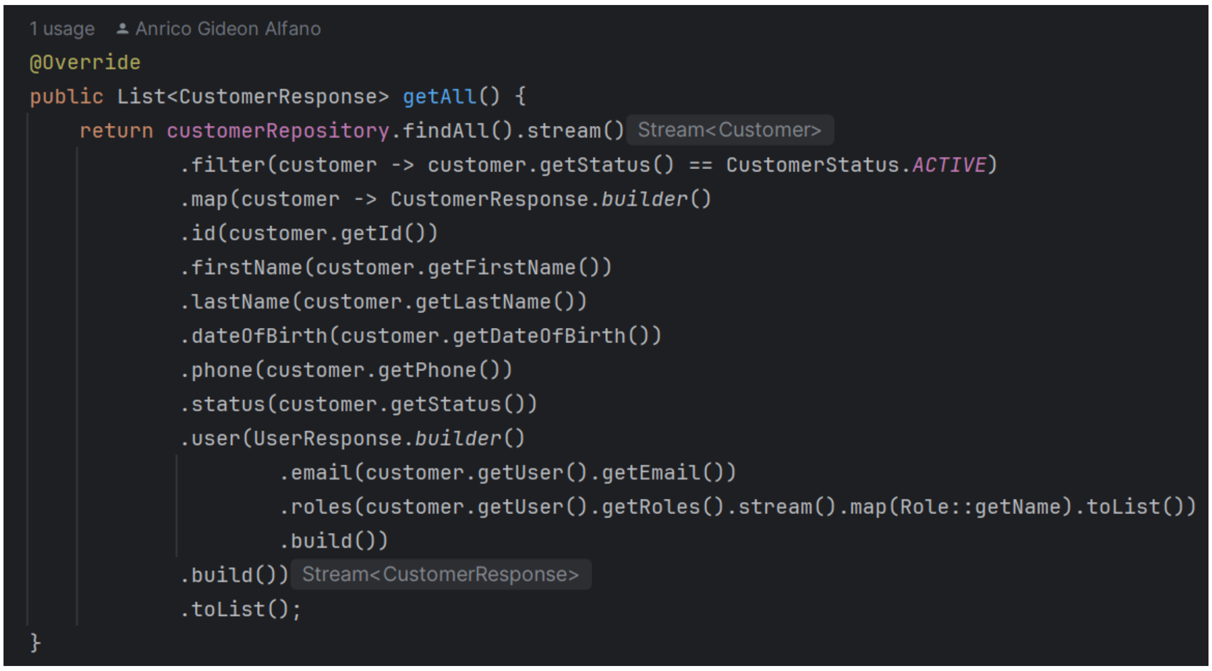
There are a number of stream usage in this api. Most of them are used on getAll methods (when displaying a list of something). A few of them are:

Stream on withClaim “role” when mapping role



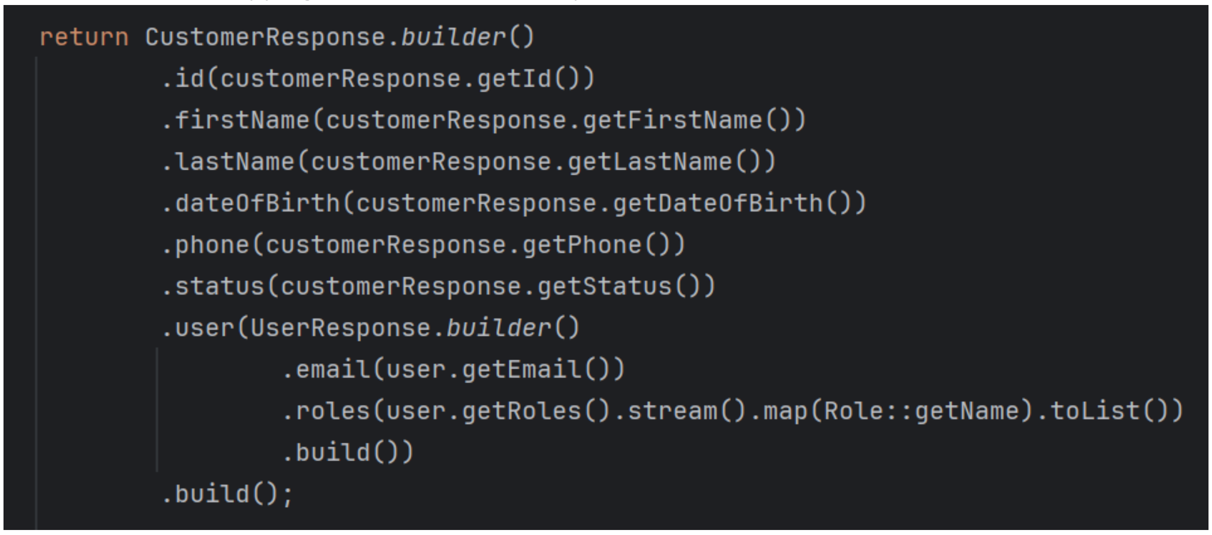
```java
1 usage   Anrico Gideon Alfano
public String generateToken(AppUser appUser) {
    try {
        Algorithm algorithm = Algorithm.HMAC256(jwtSecret.getBytes(StandardCharsets.UTF_8));
        Optional<Customer> customer = customerService.getCustomerByUserId(appUser.getUserId());
        String customerId = customer.map(Customer::getId).orElse( other: "");
        return JWT.create()
                .withIssuer(appName)
                .withSubject(appUser.getUserId())
                .withClaim( name: "customerId", customerId)
                .withExpiresAt(Instant.now().plusSeconds(jwtExpirationInSecond))
                .withIssuedAt(Instant.now())
                .withClaim( name: "role", appUser.getRoles().stream().map(Enum::name).toList())
                .sign(algorithm);
    } catch (JWTCreationException e) {
        throw new RuntimeException();
    }
}
```

Stream on displaying list of customers



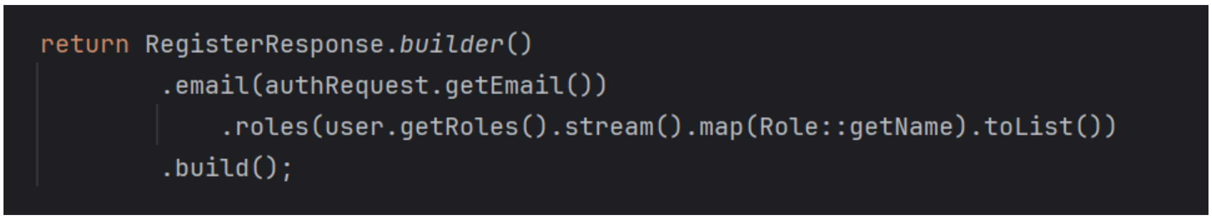
```
1 usage  Anrico Gideon Alfano
@Override
public List<CustomerResponse> getAll() {
    return customerRepository.findAll().stream() Stream<Customer>
            .filter(customer -> customer.getStatus() == CustomerStatus.ACTIVE)
            .map(customer -> CustomerResponse.builder()
            .id(customer.getId())
            .firstName(customer.getFirstName())
            .lastName(customer.getLastName())
            .dateOfBirth(customer.getDateOfBirth())
            .phone(customer.getPhone())
            .status(customer.getStatus())
            .user(UserResponse.builder()
                    .email(customer.getUser().getEmail())
                    .roles(customer.getUser().getRoles().stream().map(Role::getName).toList())
                    .build())
            .build()) Stream<CustomerResponse>
            .toList();
}
```

Stream when mapping roles in CustomerResponse



```
return CustomerResponse.builder()
        .id(customerResponse.getId())
        .firstName(customerResponse.getFirstName())
        .lastName(customerResponse.getLastName())
        .dateOfBirth(customerResponse.getDateOfBirth())
        .phone(customerResponse.getPhone())
        .status(customerResponse.getStatus())
        .user(UserResponse.builder()
                .email(user.getEmail())
                .roles(user.getRoles().stream().map(Role::getName).toList())
                .build())
        .build();
```

Stream when mapping roles in RegisterResponse



```
return RegisterResponse.builder()
        .email(authRequest.getEmail())
            .roles(user.getRoles().stream().map(Role::getName).toList())
        .build();
```