

simple Loan Management Module (LMS) which is integrated to the Bank's CORE Banking system (CBS) and the Scoring Engine.

The api endpoint **/auth/subscribe/customer** is for subscribing to LMS.

How to consume the Soap services (bank service)

The screenshot displays an IDE environment with the following components:

- Project Explorer:** Shows the project structure with folders like `constants`, `interceptor`, `model`, `repository`, `security`, and `service`. The `BankClient` folder is expanded.
- Code Editor:** Shows the `LoanController.java` file. The code includes imports for `BankClient`, `ScoringClient`, `CustomerRepository`, and `LoanBookRepository`. It also shows a `@PostMapping` method for `/transaction/data/{customerNumber}` that calls `getTransactions` from the `LoanBookRepository`.
- Endpoints Panel:** Lists the endpoints for the `auth` module. The endpoint `/bank/get/loan/status/{customerNumber}` is selected, showing its details in the right-hand pane.
- Console:** Displays the output of the application, including logs for the `ScoringEngineApplication` and the `LoanController`.

The `LoanController.java` code is as follows:

```
1 usage
2 @Autowired
3 private BankClient bankClient;
4
5 2 usages
6 @Autowired
7 private ScoringClient scoringClient;
8
9 2 usages
10 @Autowired
11 private CustomerRepository customerRepository;
12
13 3 usages
14 @Autowired
15 private LoanBookRepository loanBookRepository;
16
17 @ApiOperation(value = "get customer transaction data")
18 @PostMapping(value = "/transaction/data/{customerNumber}")
19 public ResponseEntity<CredableResponse> getTransactions(
20     @PathVariable String customerNumber
21 ) {
22 }
```

The `Endpoints` panel shows the following endpoints:

- `/bank/get/loan/status/{customerNumber}` (POST) - LoanController
- `/bank/loan/request/{customerNumber}/{amount}` (POST) - LoanController
- `/bank/transaction/data/{customerNumber}` (POST) - LoanController

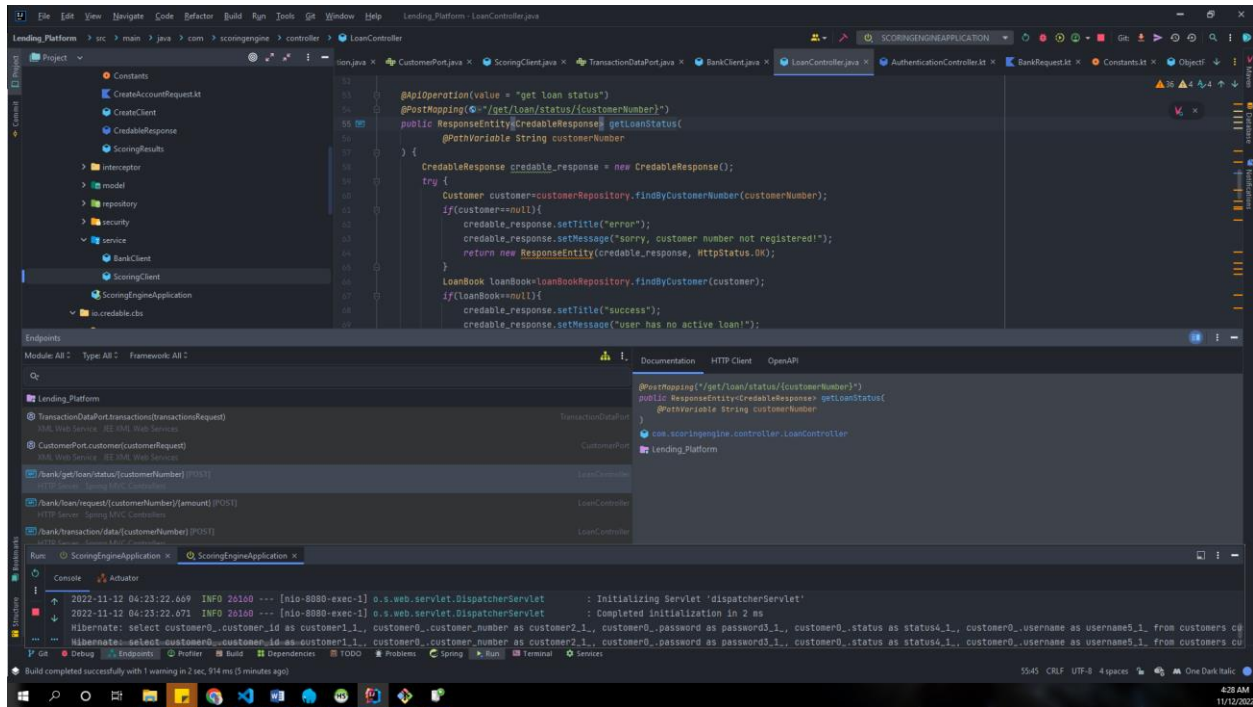
The right-hand pane shows the details for the `/bank/get/loan/status/{customerNumber}` endpoint:

```
post:
summary: "POST bank/get/loan/status/{customerNumber}"
operationId: "getLoanStatus"
parameters:
- name: "customerNumber"
in: "path"
required: true
schema:
type: "string"
responses:
"200":
```

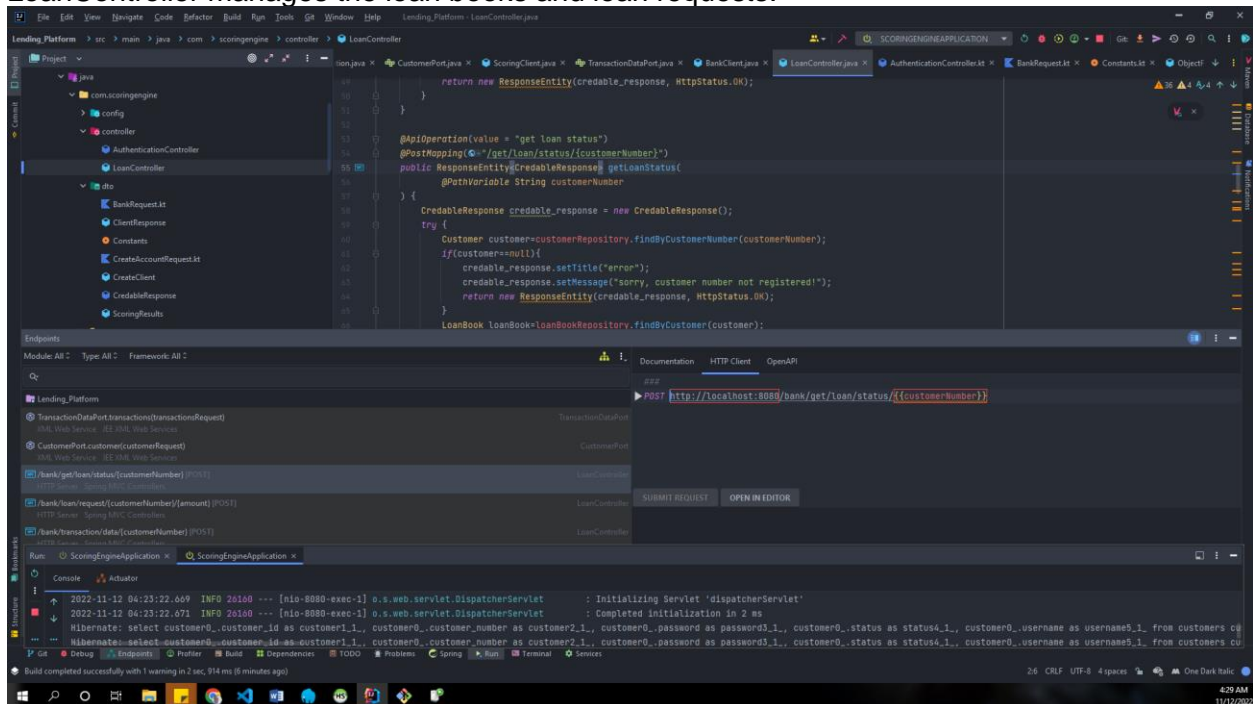
The console output shows the following logs:

```
2022-11-12 04:23:22.069 INFO 26108 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-11-12 04:23:22.071 INFO 26108 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
Hibernate: select customer0_.customer_id as customer1_1, customer0_.customer_number as customer2_1, customer0_.password as password3_1, customer0_.status as status4_1, customer0_.username as username5_1 from customers cu
```

Rest Api for Client services.



LoanController manages the loan books and loan requests.



Api Swagger Documentations

Run the application using `./mvnw spring-boot:run`

`http://127.0.0.1:8080/swagger-ui.html`

