

## Loan Application API Documentation

**API Functionality summarized:** This API enables a registered user (customer) to request a loan. The request will then be evaluated by admin and admin can APPROVE or REJECT the loan request as they see fit. If the request is approved, the customer has to pay the loan each month according to the instalment type they chose plus the interest chosen by admin. Most get all operation are only accessible by admin, adding loan type and instalment type are also admin only. Listed and documented below are only some of the feature of this API. The get service will be documented via SQL on a separate file.

### 1. Register as Customer

The screenshot shows a REST client interface for the endpoint `localhost:8080/api/auth/signup/customer` with a `POST` method. The request body is a JSON object with the following fields:

```
1 {
2   "firstName": "Sebastian",
3   "lastName": "Shaw",
4   "dateOfBirth": "1996-08-18",
5   "phone": "08871209456",
6   "email": "ss19@gmail.com",
7   "password": "ss19"
8 }
```

The response body is a JSON object with the following fields:

```
1 {
2   "message": "Successfully Registered",
3   "data": {
4     "id": "b0b89d2e-87ea-44a5-a5e6-c24de1196734",
5     "firstName": "Sebastian",
6     "lastName": "Shaw",
7     "dateOfBirth": "1996-08-18",
8     "phone": "08871209456",
9     "status": "ACTIVE",
10    "user": {
11      "email": "ss19@gmail.com",
12      "roles": [
13        "ROLE_CUSTOMER"
14      ]
15    }
16  }
17 }
```

Status: 201 Created, Time: 205 ms, Size: 687 B

### 2. Register as Admin

The screenshot shows a REST client interface for the endpoint `localhost:8080/api/auth/signup/admin` with a `POST` method. The request body is a JSON object with the following fields:

```
1 {
2   "email": "tony30@gmail.com",
3   "password": "tony30"
4 }
```

The response body is a JSON object with the following fields:

```
1 {
2   "message": "Successfully Registered as Admin",
3   "data": {
4     "email": "tony30@gmail.com",
5     "roles": [
6       "ROLE_ADMIN",
7       "ROLE_STAFF"
8     ]
9   }
10 }
```

Status: 201 Created, Time: 190 ms, Size: 546 B

### 3. Login (as admin and as customer)

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/api/auth/login`. The browser's developer tools are open, showing the Network tab with a single request. The request is a POST to `localhost:8080/api/auth/login` with a JSON body: `{ "email": "tony30@gmail.com", "password": "tony30" }`. The response is a 200 OK status with a response time of 158 ms and a size of 806 B. The response body is displayed in the console, showing a JSON object: `{ "message": "Successfully Login", "data": { "email": "tony30@gmail.com", "roles": [ "ROLE_ADMIN", "ROLE_STAFF" ], "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJhb2ZuIFRwGxpY2F0aW9uIiwiaWF0Ij0tOTBmODQyYjE6MTM3Zi00ZDZuU4U1R1Y1AtY2Y1OTY1MjMxZWZlIiwiaXNzABNjUxMjI3LChpY2Y1OTY1R3M0Q2NDc2MjcsInJvbmGUS01siUk9MRV9BRE1JT1IiOiJlJ3RlZlVlUkRBRyU1X08iLCJ1bmG0GQo-tnQrT5_C3j9m1Cf8cagH-3UxM" } }`. The browser's address bar shows the URL `http://localhost:8080/api/auth/login` and the status bar shows the response status `200 OK`.

A screenshot of a web browser's REST client interface. The top bar shows the URL 'localhost:8080/api/auth/login' and a 'POST' method. The 'Send' button is highlighted. Below the URL bar, the 'Body' tab is selected, showing a JSON payload: {"email": "ss19@gmail.com", "password": "ss19"}. The response status is '200 OK' with '117 ms' and '781 B'. The 'Pretty' tab is selected, showing the JSON response: {"message": "Successfully Login", "data": {"email": "ss19@gmail.com", "roles": ["ROLE\_CUSTOMER"], "token": "eyJ3bG6c10i1JUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3M0IjM6M2FuIFEFwGxpY2F8aW9uIiwiaWF0Ij01NTZiZTg4MjgtYjYyMy90OTBTLk1lYWUtMzFkYkE4OGExM2I2IiwiaXNwaWJoxNzA0NjUxMzExLjCjYjY0IjE3MDQ2NCM3TESlbnJvbGUiOi1Uk9MRV90VWNUOT9lU1ZldQ.1\_sWmInJbQNEZSkzskPWGv0t4yDC1rFhZph-ZNCxWkLI E"}}. The interface includes buttons for 'Save', 'Send', 'Pretty', and 'JSON'.

#### 4. Add Loan Type (admin only)

The screenshot shows the Postman interface for a REST client. The top bar indicates the current environment is 'Mandiri\_Loan\_App / Loan Type' and the action is 'Add Loan Type'. The request is a POST to 'localhost:8080/api/loan-types'. The request body is a JSON object: 

```
{  "type": "Pinjaman Kredit Bisnis",  "maxLoan": 6000000}
```

. The response is a 201 Created status, with a response time of 29 ms and a size of 579 B. The response body is a JSON object: 

```
{  "message": "Successfully added new Loan Type",  "data": {    "id": "226d8b01-daaa-46a9-923b-77f7f23d9572",    "type": "Pinjaman Kredit Bisnis",    "maxLoan": 6000000.0  },  "status": "success"}
```

## 5. Add Instalment Type (admin only)

The screenshot shows the Postman interface for a REST client. At the top, the breadcrumb navigation indicates the path: Mandiri\_Loan\_App / Installment Type / Add New Installment Type (Admin & Staff Only). The request method is set to POST, and the URL is localhost:8080/api/installment-types. The request body is a JSON object: {"installmentType": "SIXTH\_MONTHS"}. The response status is 201 Created, with a time of 332 ms and a size of 567 B. The response body is a JSON object: {"message": "Successfully added new Installment Type", "data": {"id": "fd86c3dd-0451-46bd-b23a-f85e69618f41", "installmentType": "SIXTH\_MONTHS"}}.

## 6. Request a Loan (customer only)

Can be seen null value will be updated when the request is approved by admin.

Loan\_App / Transaction / Create Transaction

POST localhost:8080/api/transactions/approve

Params Authorization Headers (10) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary

```
1 {
2   "loanType": {
3     "id": "afcce087-baa2-4508-a712-f6cfc8fa59df"
4   },
5   "installmentType": {
6     "id": "b9795e00-5071-4926-bd2f-83c01233ea85"
7   },
8   "customer": {
9     "id": "d99ec12f-213a-4a60-aba5-b2504430e244"
10  },
11  "nominal": 8000000
12 }
```

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Loan Request Has Been Saved",
3   "data": {
4     "id": "63dc9c23-f355-4afc-80a9-a4c1046a5d8d",
5     "loanType": {
6       "id": "afcce087-baa2-4508-a712-f6cfc8fa59df",
7       "type": "Pinjaman Kredit Rumah",
8       "maxLoan": 9.0E7
9     },
10    "installmentType": {
11      "id": "b9795e00-5071-4926-bd2f-83c01233ea85",
12      "installmentType": "TWELVE_MONTHS"
13    },
14    "customer": {
15      "id": "d99ec12f-213a-4a60-aba5-b2504430e244",
16      "firstName": "Michael",
17      "lastName": "Hersey",
```

```
    "dateOfBirth": "1990-10-21",
    "phone": "0890121741",
    "status": "ACTIVE",
    "user": {
      "id": "5beb0a3b-0053-484f-85f2-8e3f3bc986f6",
      "email": "mike19@gmail.com",
    }
  }
},
"nominal": 8.0E7,
"interestRate": null,
"approvedAt": null,
"approvedBy": null,
"approvalStatus": null,
"loanTransactionDetails": null,
"createdAt": 20240107,
"updatedAt": null
}
```

## 7. Approve Loan Request (admin only)

Mandiri\_Loan\_App / Transaction / Approve Loan Request

PUT localhost:8080/api/transactions/approve

Params Authorization Headers (10) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 {
2   "loanTransactionId": "63dc9c23-f355-4afc-80a9-a4c1046a5d8d",
3   "interestRates": 2,
4   "approvalStatus": 0
5 }
```

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Loan Request Approved",
3   "data": {
4     "id": "63dc9c23-f355-4afc-80a9-a4c1046a5d8d",
5     "loanType": {
6       "id": "afcce087-baa2-4508-a712-f6cfc8fa59df",
7       "type": "Pinjaman Kredit Rumah",
8       "maxLoan": 9.0E7
9     },
10    "installmentType": {
11      "id": "b9795e00-5071-4926-bd2f-83c01233ea85",
12      "installmentType": "TWELVE_MONTHS"
13    },
14    "customer": {
15      "id": "d99ec12f-213a-4a60-aba5-b2504430e244",
16      "firstName": "Michael",
17      "lastName": "Hersey",
18      "dateOfBirth": "1990-10-21",
19      "phone": "0890121741",
20      "status": "ACTIVE",
21      "user": {
22        "id": "5beb0a3b-0053-484f-85f2-8e3f3bc986f6",
23        "email": "mike19@gmail.com",
```

```
    "roles": [
      {
        "role_id": "8f96eb81-d235-45d5-8b52-3d91679dbe33",
        "name": "ROLE_CUSTOMER"
      }
    ],
    "nominal": 8.0E7,
    "interestRate": 2,
    "approvedAt": 20240108,
    "approvedBy": "tony38@gmail.com",
    "approvalStatus": "APPROVED",
    "loanTransactionDetails": [
      {
        "id": "98b90366-0398-4ab8-9450-6a09abdb3693",
        "transactionDate": 20240108,
        "nominal": 8.16E7,
        "loanStatus": "UNPAID",
        "updatedAt": 20240108
      }
    ],
    "createdAt": 20240107,
    "updatedAt": 20240108
  }
}
```

## 8. Pay Instalment (customer only)

### a. When paying (not fully paid)

The screenshot shows a REST client interface for a PUT request to `localhost:8080/api/transactions/1ae7999c-a176-4ac6-8c73-c5b5eea840b4/pay`. The 'Body' tab is selected, displaying a JSON response in 'Pretty' format:

```
1 {
2   "message": "Payment Successful",
3   "data": "LoanStatus: UNPAID, Remaining: 3366666.666666667"
4 }
```

### b. When fully paid

The screenshot shows a REST client interface for a PUT request to `localhost:8080/api/transactions/1ae7999c-a176-4ac6-8c73-c5b5eea840b4/pay`. The 'Body' tab is selected, displaying a JSON response in 'Pretty' format:

```
1 {
2   "message": "Payment Successful",
3   "data": "LoanStatus: PAID, Remaining: 0.0"
4 }
```

### c. When trying to pay an already paid loan

The screenshot shows a REST client interface for a PUT request to `localhost:8080/api/transactions/4a63888a-91cf-420e-b5c7-f36e278aced/pay`. The 'Body' tab is selected, displaying a JSON response in 'Pretty' format:

```
1 {
2   "message": "Your loan is already paid off. Payment not executed.",
3   "data": "PAID"
4 }
```