**Bank-Web-App**

**Schedule Service**

TABLE OF CONTENTS

[1.INTRODUCTION 2](#_Toc204787340)

[2. TECH STACK 3](#_Toc204787341)

[3. DATA DICTIONARY 4](#_Toc204787342)

[3.1.1 accounts 5](#_Toc204787343)

[3.1.2 LoanPayment 5](#_Toc204787344)

[4.SCHADULES 6](#_Toc204787345)

[4.1.1 EMI Deduction Schedule 6](#_Toc204787346)

[4.1.2 Loan Payment Reminder Email Schedule 7](#_Toc204787347)

[4.1.3 Monthly Transaction Email Schedule (User Groups) 8](#_Toc204787348)

[5.PROJECT STRACTURE 9](#_Toc204787349)

# 1.INTRODUCTION

The **Schedule Service** is responsible for automating monthly tasks related to loan payments. It plays a crucial role in ensuring timely payment initiation, customer reminders, and email notifications. This service runs on a scheduled basis—typically at the **start of each month**—to handle routine operations efficiently and seamlessly integrate with other services in the ecosystem.

**Features**

**Monthly Loan Payment Initiation**

* Automatically triggers loan payment requests by calling the **Transaction Service**.
* Ensures all active loan accounts are processed on their scheduled due date (e.g., the 1st of each month).

**Payment Reminder Notification**

* Initiates a call to the **Notification Service** to send reminders to customers regarding upcoming or due payments.
* Supports multiple channels (e.g., email, SMS, push notifications).

**Transaction Email Dispatch**

* Sends monthly transaction summaries or receipts to customers.
* Email is triggered only after the successful initiation of the payment transaction.

**Communication**

* Kafka
* FeignClient
* RestTemplate

**Service Dependency**

* Transection Service
* Notification Service

# 2. TECH STACK

|  |  |
| --- | --- |
| Technology | Version |
| JDK | 17 |
| Maven | 3.6.3 |
| Spring Boot | 3.5.3 |
| MongoDB | - |
| Kafka | 7.4.3 |
| Zookeeper | 7.4.3 |
| Zipkin | 2.23 |

# 3. DATA DICTIONARY

DB: **LoanPayment**

|  |  |  |
| --- | --- | --- |
| No | Table Name | Description |
| 1 | accounts | Stores user account details such as account number, branch, and account type. |
| 2 | loan\_payment | Stores scheduled loan payments, EMI amounts, and reminder status. |

## 3.1.1 accounts

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Sample Value |
| id | String | Unique identifier for the account (MongoDB \_id) | "64e41f70b1f3c2553e123abc" |
| accountNum | Long | Unique numeric account number assigned to the user | 100002345678 |
| branchId | String | Identifier for the branch where the account is held | "BR001" |
| type | String | Type of account (e.g., savings, current, loan) | "savings" |

## 3.1.2 LoanPayment

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Sample Value |
| id | String | Unique identifier for the loan payment document (MongoDB \_id) | "64e41f70b1f3c2553e123abc" |
| loanId | Long | Unique identifier for the loan | 1234567890 |
| paymentId | Long | Unique identifier for the associated payment | 9876543210 |
| scheduleDate | String | Scheduled date for the EMI payment (format: YYYY-MM-DD) | "2025-08-01" |
| emi | Double | Monthly installment amount to be paid | 15000.00 |
| balance | Double | Remaining loan balance after payment | 250000.00 |
| isReminderSend | Boolean | Flag indicating if a reminder notification has been sent | true |
| email | String | Customer’s email address for payment notifications | "user@example.com" |

# 4.SCHADULES

## 4.1.1 EMI Deduction Schedule

|  |  |  |
| --- | --- | --- |
| Field | Description | Value / Example |
| Schedule Type | Type of scheduled task | EMI Deduction |
| Cron Expression | Schedule time in cron format | 0 0 0 12 \* \* |
| Run Time | Human-readable schedule time | Every day at noon (12 AM) |
| Action Performed | What the schedule triggers | Deduct EMI payment from user account |
| Related Service | Service called to perform the action | Transaction Service |
| Frequency | How often the task runs | Daily |
| Notes | Additional info | Deduction happens for all active loans payments |

## 4.1.2 Loan Payment Reminder Email Schedule

|  |  |  |
| --- | --- | --- |
| Field | Description | Value / Example |
| Schedule Type | Type of scheduled task | Loan Payment Reminder Email |
| Cron Expression | Schedule time in cron format | 0 0 14 \* \* \* |
| Run Time | Human-readable schedule time | Every day at 2:00 PM (14:00) |
| Action Performed | Sends reminder emails | Send email reminder 2 days before payment due date |
| Related Service | Service responsible for sending emails | Notification Service |
| Frequency | How often the task runs | Daily |
| Notes | Additional information | Checks loans with payment dates 2 days ahead and sends reminders |

## 4.1.3 Monthly Transaction Email Schedule (User Groups)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group | Cron Expression | Run Time | User Range | Description |
| Group 1 | 0 0 9 1 \* ? | 1st of every month, 9:00 AM | 0% – 25% | Sends transaction emails to the first 25% of users |
| Group 2 | 0 0 10 1 \* ? | 1st of every month, 10:00 AM | 26% – 50% | Sends transaction emails to the next 25% of users |
| Group 3 | 0 0 11 1 \* ? | 1st of every month, 11:00 AM | 51% – 75% | Sends transaction emails to the third 25% of users |
| Group 4 | 0 0 12 1 \* ? | 1st of every month, 12:00 PM | 76% – 100% | Sends transaction emails to the final 25% of users |

**Purpose**

This staggered scheduling helps:

* Avoid system overload
* Optimize email delivery performance
* Balance transactional load across the day

## 5.PROJECT STRACTURE

schedule-service/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ └── com/

│ │ │ └── com.bank.web.schedule./

│ │ │ ├── config/

│ │ │ ├── service/

│ │ │ ├── repo/

│ │ │ ├── model/

│ │ │ ├── dto/

│ │ │ ├── kafka/

│ │ │ └── ScheduleApplication.java

│ └── resources/

│ ├── application.yml

│ ├── static/

│ └── templates/

├── test/

│ └── java/

│ └── com/

│ └── com.bank.web.schedule/