**CRM Microservices API Specification**

**1. Overview**

The CRM microservices provide RESTful APIs for handling customer service requests submitted by bank staff via Call Center or Branch. This API allows for:

* Creating customer service requests
* Retrieving existing requests
* Updating request statuses
* Sending and receiving events via Kafka (if applicable)

**2. Base URL**

http://{host}:{port}/api/requests

**3. Authentication**

* Future enhancement: Implement authentication via **Spring Security**

A close up of a check

AI-generated content may be incorrect.

**4. API Endpoints**

**4.1 Create Service Request**

**Endpoint:**

POST /api/v1/requests

**Description:** Creates a new service request.

**Request Body:**

A screenshot of a document

AI-generated content may be incorrect.

{

    "customerId": "CUST121",

    "type": "CREDIT\_CARD\_ISSUE",

    "description": "Customer reported failed CREDIT\_CARD\_ISSUE transaction",

    "priority": "HIGH",

    "assignedDepartment": "PAYMENTS\_DEPT",

    "metadata": {

        "transactionId": "TRX780",

        "amount": 100000.00,

        "currency": "THB",

        "paymentMethod": "CREDIT\_CARD",

        "errorCode": "PAY\_001"

    }

}

**Response:**

* **200 OK: Returns the list of requests.**
* **400 Bad Request**: Invalid input data.
* **401 Unauthorized** The request is unauthenticated.

**Response Body:**

{

    "id": 8,

    "customerId": "CUST100",

    "type": "CREDIT\_CARD\_ISSUE",

    "description": "Customer reported failed CREDIT\_CARD\_ISSUE transaction",

    "status": "NEW",

    "priority": "HIGH",

    "assignedDepartment": **null**,

    "createdAt": "2025-02-16T21:46:35.7918999",

    "updatedAt": **null**,

    "metadata": {

        "transactionId": "TRX780",

        "amount": 100000.0,

        "currency": "THB",

        "paymentMethod": "CREDIT\_CARD",

        "errorCode": "PAY\_001"

    }

}

**4.2 Retrieve All Requests**

**Endpoint:**

GET /api/requests

**Description:** Retrieves all customer service requests.

**Response:**

* **200 OK**: Returns the list of requests.

**Response Body:**

[

{

"id": 1,

"customerName": "John Doe",

"requestType": "Credit Card Payment Inquiry",

"requestDetails": "Check remaining balance on credit card",

"status": "RECEIVED"

},

{

"id": 2,

"customerName": "Jane Smith",

"requestType": "Account Balance Inquiry",

"requestDetails": "Check remaining balance in savings account",

"status": "IN\_PROGRESS"

}

]

**4.3 Update Request Status**

**Endpoint:**

PATCH /api/requests/{id}/status?status={status}

**Description:** Updates the status of a specific request.

**Path Parameter:**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| id | Long | Yes | The unique ID of the request |

**Query Parameter:**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| status | Enum | Yes | The new status (RECEIVED, IN\_PROGRESS, COMPLETED, CANCELED) |

**Response:**

* **200 OK**: Request status successfully updated.
* **404 Not Found**: Request not found.

**Response Body:**

{

"id": 1,

"customerName": "John Doe",

"requestType": "Credit Card Payment Inquiry",

"requestDetails": "Check remaining balance on credit card",

"status": "IN\_PROGRESS"

}

**4.4 Send Event to Kafka (For Microservice Integration)**

**Endpoint:**

POST /api/events

**Description:** Sends a request event to Kafka.

**Request Body:**

{

"id": 1,

"customerName": "John Doe",

"requestType": "Credit Card Payment Inquiry",

"requestDetails": "Check remaining balance on credit card",

"status": "IN\_PROGRESS"

}

**Response:**

* **200 OK**: Event successfully sent.

**4.5 Receive Event from Kafka**

**Kafka Topic:** request-topic

**Consumer Service:**

* Listens for request status updates and processes them.
* Updates request statuses based on event messages received.

**Sample Event Message:**

{

"id": 1,

"status": "COMPLETED"

}

**5. Error Handling**

| **HTTP Status Code** | **Meaning** |
| --- | --- |
| 400 Bad Request | Invalid input data |
| 404 Not Found | Requested resource not found |
| 500 Internal Server Error | Unexpected server error |

**6. Future Enhancements**

* Implement authentication via **Keycloak**.
* Store request history for audit purposes.
* Add pagination and filtering for GET requests.
* Deploy with **Docker & Kubernetes**.