|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Date | Version | issuerer | Details |
| 28.04.2024 | 1.0.0 | Ozgur Yatmaz | SecurePay Payment service API is created. |
| 29.04.2024 | 1.0.0 | Ozgur Yatmaz | API document is created. |

**Index**

[1 Summary 3](#__RefHeading___Toc3957_1262165384)

[2 API Operations 3](#__RefHeading___Toc11629_1023073916)

[2.1. Add Customer Operation 3](#__RefHeading___Toc13755_1543940018)

[2.1.2. Add Customer Request 4](#__RefHeading___Toc73825_1966934515)

[2.1.4. Add Customer Response 5](#__RefHeading___Toc73827_1966934515)

[2.2. Get All Customers Operation 5](#__RefHeading___Toc7773_349496368)

[2.3. Make Payment Operation 6](#__RefHeading___Toc3961_1262165384)

[2.4. Query Payments by card Number or customer Number operation 8](#__RefHeading___Toc3963_1262165384)

[2.5. Query Payments by date interwal 9](#__RefHeading___Toc3997_1262165384)

# **1 Summary**

This api anables saving customers and cards of the customers to mysql database and requesting payments to any external payment service providers. And if external payment service confirms that the payment is made card balance is updated and payment is recorded to payments table. Lastly, all payments can be queried by date interval or curtomer number or card number.

Besides; api is error proof. All of the possible failure cases are considered and handled properly.

Further more, Spring security can be added if requested. As it is fundamental part of any payment system such that payment request can be processed if and only if correct card credentials are provided.

# 2 **API Operations**

## **2.1. Add Customer Operation**

Customer and all associated cards are saved to data center with this operation.

| **Request** | | | |
| --- | --- | --- | --- |
| Fields of Customer Object | | | |
| **Field Name** | **Data Type** | **Required** | **Detail** |
| name | String | Yes | Customer name |
| email | String | Yes | Email of the customer (must be unique) |
| customerNumber | String | Yes | Identification number of the customer (must be unique) |
| cards | List<Card> | No | List of cards that will be associated with the customer. If it is not provided a default card of balance zero will be auto created. |

| **Request** | | | |
| --- | --- | --- | --- |
| Fields of Card Object | | | |
| **Field Name** | **Data Type** | **Required** | **Detail** |
| cardNumber | String | Yes | Card number (must be unique) |
| balance | double | no | Amount of money in card |

### **2.1.2. Add Customer Request**

| **Add Customer Request REST** |
| --- |
| **Post Request:** http://localhost:8080/customer/add-customer |
| {  "name": "Ozgur Yatmaz",  "email": "o5.y@firisbe.com",  "customerNumber": "114-5",  "cards": [  {  "cardNumber": "571-1",  "balance": 1000.00  },  {  "cardNumber": "571-2",  "balance": 500.80  }  ]  } |

### **2.1.4. Add Customer Response**

| **Add Customer Responses** | |
| --- | --- |
| **Rest -** | |
| SUCCESS | Status code 201 (created) with text message: Customer with id 6 is created: |
| FAILED | {  "timestamp": "2024-04-28T23:42:19.9544279",  "message": "Total errors: 2 and first error is: Customer name may not be empty or blank",  "details": "uri=/customer/add-customer"  } |
| {  "timestamp": "2024-04-28T23:44:33.1666515",  "message": "Conflict in provided data exception",  "details": "Customer number 114-1 is already allocated"  } |
| {  "timestamp": "2024-04-28T23:46:27.3711533",  "message": "Conflict in provided data exception",  "details": "Following card numbers are already used: 571-1, 571-2, "  } |

## **2.2. Get All Customers Operation**

All customers in data center is fetched with this operation. Response is list of customer object without card info. Content of the customer class is described in the table above.

| **Get All Customers Responses** | |
| --- | --- |
| **Get Request** http://localhost:8080/customer/get-all-customers | |
| SUCCESS | [  {  "name": "ozgur yatmaz",  "customerNumber": "114-1",  "email": "o.y@firisby.com"  },  {  "name": "ozgur2 yatmaz2",  "customerNumber": "114-2",  "email": "o2.y@firisby.com"  }  ] |
| FAILED | {  "timestamp": "2024-04-29T02:29:20.3095335",  "message": "Error occurred while retrieving customers!",  "details": "/ by zero"  } |

## **2.3. Make Payment Operation**

Provides to make payment from the balance of the card.

| **Request** | | | |
| --- | --- | --- | --- |
| Fields of Payment Request Object | | | |
| **Field Name** | **Data Type** | **Required** | **Detail** |
| cardNumber | String | Yes | Card identification number of the associated payment. |
| amount | double | Yes | Amount to be paid. |

| **Make Payment Request** |
| --- |
| **Post Request:** http://localhost:8080/payments/make-payment |
| {  "cardNumber": "571-1",  "amount": 1.0  } |

| **Make Payment Responses** | |
| --- | --- |
| **Rest -** | |
| SUCCESS | Status code 200 (Ok) with text message: Payment is made |
| FAILED | {  "timestamp": "2024-04-29T02:47:10.2696662",  "message": "Total errors: 1 and first error is: Payment amount should be positive:) ",  "details": "uri=/payments/make-payment"  } |
| {  "timestamp": "2024-04-29T02:48:52.9508228",  "message": "Card not found",  "details": "Card number 571-111 is not associated with any customer"  } |
| {  "timestamp": "2024-04-29T02:50:12.9474079",  "message": "Error while sending request to external payment service provider",  "details":"I/O error on POST request for http://localhost:3004/mock-service/make-payment: Connection refused: no further information"  } |
| {  "timestamp": "2024-04-29T02:52:58.4958217",  "message": "Response Code: 102 PROCESSING",  "details": "Payment failed!"  } |

## **2.4. Query Payments by card Number or customer Number operation**

Return all payments associated with provided customer number or card number. İt is a get request with two optional parameters.

| **Qery Payments Response** | | | |
| --- | --- | --- | --- |
| Fields of Payment Object (returned in a list) | | | |
| **Field Name** | **Data Type** | **Required** | **Detail** |
| amount | double |  | Paid amount |
| customerNumber | String |  | Customer number of the asociated payment |
| cardNumber | String |  | Card number of the asociated payment |
| paymentDate | LocalDateTime |  | Payment time |

| **Get All Customers Request and Responses** | |
| --- | --- |
| <http://localhost:8080/payments/fetch-payments?customerNumber=114-1> | |
| <http://localhost:8080/payments/fetch-payments?cardNumber=571-1> | |
| http://localhost:8080/payments/fetch-payments?cardNumber=571-1&customerNumber=114-1 | |
| SUCCESS | [  {  "amount": 62.57,  "cardNumber": "571-1",  "customerNumber": "114-1",  "paymentDate": "2024-04-26T01:36:09.759075"  },  {  "amount": 62.58,  "cardNumber": "571-1",  "customerNumber": "114-1",  "paymentDate": "2024-04-26T01:36:17.452867"  },  {  "amount": 62.59,  "cardNumber": "571-2",  "customerNumber": "114-1",  "paymentDate": "2024-04-26T01:36:34.42784"  }  ] |
| FAILED | {  "timestamp": "2024-04-29T02:29:20.3095335",  "message": "Error occurred while retrieving customers!",  "details": "/ by zero"  } |

## **2.5. Query Payments by date interwal**

Return all payments associated with provided date interval. İt is a get request with two compulsory parameters.

| **Get All Customers by date interval Request and Responses** |
| --- |
| http://localhost:8080/payments/fetch-payments-bydate?startDate=2024-04-28&endDate=2024-04-29 |
| **Fail case if any of the parameters are not provided:** |
| {  "type": "about:blank",  "title": "Bad Request",  "status": 400,  "detail": "Required parameter 'endDate' is not present.",  "instance": "/payments/fetch-payments-bydate"  } |