

## ValidationService

```
9 usages Kevin Octavianus Halim
@Service
public class ValidationService {
    2 usages
    private Validator validator;

    no usages Kevin Octavianus Halim
    public ValidationService(Validator validator) { this.validator = validator; }

    7 usages Kevin Octavianus Halim
    public List<String> validate(Object request) {
        Set<ConstraintViolation<Object>> constraintViolations = validator.validate(request);
        List<String> violationList = new ArrayList<>();
        if (constraintViolations.size() != 0) {
            for(ConstraintViolation c : constraintViolations){
                violationList.add(c.getMessage());
            }
            return violationList;
        }
        return null;
    }
}
```

Method `validate()` berguna untuk mengumpulkan violations yang didapatkan saat terjadi adanya pelanggaran constraints yang ada pada `RequestBody`, yang saya gunakan biasanya untuk Business Logic yang membutuhkan Input dari User seperti saat Register, Create Product, dll.

Salah Satu Contoh penggunaan `ValidationService.validate()` ada pada `MerchantServiceImpl createProduct()`

```
@Transactional
@Override
public ControllerResponse<?> createProduct(ProductCreateRequest request, String merchantId)
    try{
        List<String> violations = validation.validate(request);
        if(violations != null) {
            ControllerResponse<List<String>> response = new ControllerResponse<>();
            response.setStatusCode(HttpStatus.BAD_REQUEST.value());
            response.setMessage(HttpStatus.BAD_REQUEST.getReasonPhrase());
            response.setData(violations);
            return response;
        }
    }
```

List yang berupa message violations akan di return menjadi response yang nantinya akan dikirimkan ke Controller, sehingga list violations terlihat pada response body.

Note: penggunaan try catch adalah untuk menangkap jika ada eror yang tidak saya duga, dan dapat menampilkan eror tersebut pada response controller untuk menandakan bahwa service terdapat error sehingga mempermudah proses debugging.

## Redeem Rewards

POST

/api/rewards/redeem

Redeem A Reward CUSTOMER OR SUPER\_ADMIN ONLY

Redeem A Rewards Based On Available Rewards, CUSTOMER OR SUPER\_ADMIN ONLY

Parameters

Try it out

Name	Description
<b>customerId</b> * required string (query)	<input type="text" value="customerId"/>
<b>rewardId</b> * required string (query)	<input type="text" value="rewardId"/>

Invalid ID / Not Found

Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Customer Not Found / Invalid Customer Id" }</pre></div> <div><div></div>Download</div>

Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Reward Not Found / Invalid Reward Id" }</pre></div> <div><div></div>Download</div>

Insufficient Points




Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 406,   "message": "Not Acceptable",   "data": "Insufficient Customer's Points / Customer's Points Is Too Low To Redeem This Reward" }</pre></div> <div><div></div>Download</div>

## Register As Customer

POST

/auth/register-as-customer

Register As Customer



Register As A Customer and Gain Role AS CUSTOMER

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value

Schema

```
{
  "name": "Kevin Halim",
  "phoneNumber": "+62899012345",
  "birthDate": "2000-12-31",
  "email": "customer123@gmail.com",
  "password": "Customer123"
}
```

### Bad Request

Request body

required

application/json

```
{
}
```


Code

Details

200

Response body

```
{
  "statusCode": 400,
  "message": "Bad Request",
  "data": [
    "Phone number must not be blank",
    "Email must Not be Blank",
    "Name must not be blank",
    "Birth Date Must Not Be Empty",
    "Birth date must not be null"
  ]
}
```



Download

Request body

required

application/json

```
{
  "name": "Kevin Halim",
  "phoneNumber": "0789884243",
  "birthDate": "1998 Oktober ",
  "email": "customer123",
  "password": "customer123"
}
```


Code

Details

200

Response body

```
{
  "statusCode": 400,
  "message": "Bad Request",
  "data": [
    "Password Must Contain at least 1 lowercase letter, 1 uppercase letter, and 1 number",
    "Email must be valid and contain only lowercase letters and numbers.",
    "Phone number must begin with +62 and have 10 to 14 additional digits"
  ]
}
```



Download

## Login

**POST** `/auth/login` Login  

Login to Get Token

Parameters

Try it out

No parameters


Request body required

application/json 


Example Value | Schema

```
{
  "email": "superadmin@gmail.com",
  "password": "superadmin123"
}
```

### Invalid Combination

Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 403,   "message": "Forbidden",   "data": "Bad credentials" }</pre></div> <div> <b>Download</b></div>

### Bad Request



Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 400,   "message": "Bad Request",   "data": [     "email must not be empty",     "password must not be empty"   ] }</pre></div> <div> <b>Download</b></div>

### View Merchant's Product List

GET

/api/merchants/{merchantId}/products

View Product List



View Product List by It's Merchant Id

Parameters

Try it out

Name	Description
page integer(\$int32) (query)	Default value : 0 <input type="text" value="0"/>
size integer(\$int32) (query)	Default value : 5 <input type="text" value="5"/>
merchantId * required string (path)	<input type="text" value="merchantId"/>

Merchant Not Found / Invalid Id

merchantId \* required

string  
*(path)*

qwerty12345

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/merchants/qwerty12345/products?page=0&size=5' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJpdWp1TXVudHVyQVBGIiwic3ViIjoiazDlmcWE3YzAtNmZiMC00MGNhbnR5cyJ9.eyJpc3MiOiJpdWp1TXVudHVyQVBGIiwic3ViIjoiazDlmcWE3YzAtNmZiMC00MGNhbnR5cyJ9'
```

Request URL

```
http://localhost:8080/api/merchants/qwerty12345/products?page=0&size=5
```

Server response

Code	Details
200	<div>Response body<pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Merchant Not Found / Invalid Merchant Id" }</pre></div> <div><div>Download</div></div>

## Create Transaction Validation

POST

/api/transactions

Create A Transaction CUSTOMER OR SUPER\_ADMIN ONLY

Create New Transaction, CUSTOMER OR SUPER\_ADMIN ONLY !!

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value

Schema

```
{
  "merchantId": "string",
  "customerId": "string",
  "productId": "string"
}
```

## Not Authorized / Forbidden

Code	Details
403 <i>Undocumented</i>	<div>Error: response status is 403</div> <div>Response body</div> <div><pre>{   "timestamp": "2024-05-12T06:28:16.548+00:00",   "status": 403,   "error": "Forbidden",   "path": "/api/transactions" }</pre></div> <div><div>Download</div></div>

## Bad Request

Request body

required

application/json

```
{
}
```

Code

Details

200

Response body

```
{
  "statusCode": 400,
  "message": "Bad Request",
  "data": [
    "Merchant Id must not be blank",
    "Product Id must not be blank",
    "Customer Id must not be blank"
  ]
}
```

Download

## Authorized But Invalid Id

Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Merchant Not Found / Invalid Merchant Id" }</pre></div> <div> <a href="#">Download</a></div>
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Product Not Found / Invalid Product Id" }</pre></div> <div> <a href="#">Download</a></div>
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Customer Not Found / Invalid Customer Id" }</pre></div> <div> <a href="#">Download</a></div>

## View Customer's Details



### Customer Controller

Controller To Get Customer Details

GET

/api/customers/{customerId}

Get Customer's Details


Get A Customer's Details

Parameters

Try it out

Name	Description
<b>customerId</b> * required string (path)	<input type="text" value="customerId"/>

## Invalid Id

Code	Details
200	<div>Response body</div> <div><pre>{   "statusCode": 404,   "message": "Not Found",   "data": "Customer Not Found / Invalid Customer Id" }</pre></div> <div> <a href="#">Download</a></div>