

POST TRANSACTION INCOME:

Request

URI: /transaction-income

Method: POST

Body:

```
{  
    "name": TransactionIncomeType – "ENUM",  
    "description": String – "string",  
    "amount": BigDecimal(18,2) – 00.00,  
    "date": LocalDate – "0000-00-00"  
}
```

Response

1. Success - OK

Code: 201 – Created

Header: Location – URI of created resource

2. Validation Error - OK

Code: 422 – Unprocessable Entity

Body:

```
{  
    "status": 422,  
    "message":  
        "Transaction income name cannot be null",  
        "Transaction income amount cannot be null",  
        "Transaction income date cannot be null"  
    "errors": []  
}
```

3. Duplicated Transaction - OK

Code: 409 – Conflict

Body:

```
{  
    "status": 409,  
    "message": "Transaction income already exists",  
    "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

5. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionIncomeType: +name

6. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{  
    "status": 409,  
    "message":  
        "Transaction amount must be greater than zero.",  
        "Transaction amount exceed the allowed limit of  
        9999999999999999.99.",  
        "Transaction amount must have at most two decimal places."  
    "errors": []  
}
```

GET TRANSACTION INCOME

Request

URI: /transaction-income/{id}

Method: GET

Response

1. Success - OK

Code 200 – OK

Body:

```
{  
  "id": UUID – "0000-ffff-0000-ffff",  
  "name": TransactionIncomeType – "ENUM",  
  "description": String – "string",  
  "amount": BigDecimal(18,2) – 00.00,  
  "date": LocalDate – "0000-00-00",  
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",  
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",  
  "user_id": UUID – "0000-ffff-0000-ffff"  
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid UUID Format",  
  "errors": []  
}
```

Request

URI: /transaction-income

Query Params: name, description, amount, date or no params to get all

Method: GET

Response:

1. Success

Code 200 – OK

Body:

```
{  
  "id": UUID – "0000-ffff-0000-ffff",  
  "name": TransactionIncomeType – "ENUM",  
  "description": String – "string",  
  "amount": BigDecimal(18,2) – 00.00,  
  "date": LocalDate – "0000-00-00",  
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",  
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",  
  "user_id": UUID – "0000-ffff-0000-ffff"  
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Name - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid value Provided for TransactionIncomeType: +name",  
  "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD

5. End date before Start date

Code: 400 – Bad Request

Body

```
{  
    "status": 400,  
    "message": "Start date cannot be after end date.",  
    "errors": []  
}
```

PUT TRANSACTION INCOME

Request

URI: /transaction-income/{id}

Method: PUT

Response

1. Success - OK

Code 204 – No Content

2. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

3. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionIncomeType: +name

4. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 409,  
  "message":  
    "Transaction amount must be greater than zero.",  
    "Transaction amount exceed the allowed limit of  
    9999999999999999.99.",  
    "Transaction amount must have at most two decimal places."  
  "errors": []  
}
```

DELETE TRANSACTION INCOME

Request

URI: /transaction-income/{id}

Method: DELETE

Response

1. Success - OK

Code 204 – No Content

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid UUID string: + id",  
  "errors": []  
}
```

POST TRANSACTION BENEFIT:

Request

URI: /transaction-benefit

Method: POST

Body:

```
{  
    "name": TransactionBenefitType – "ENUM",  
    "description": String – "string",  
    "amount": BigDecimal(18,2) – 00.00,  
    "date": LocalDate – "0000-00-00"  
}
```

Response

1. Success - OK

Code: 201 – Created

Header: Location – URI of created resource

2. Validation Error - OK

Code: 422 – Unprocessable Entity

Body:

```
{  
    "status": 422,  
    "message":  
        "Transaction benefit name cannot be null",  
        "Transaction benefit amount cannot be null",  
        "Transaction benefit date cannot be null"  
    "errors": []  
}
```

3. Duplicated Transaction - OK

Code: 409 – Conflict

Body:

```
{  
    "status": 409,  
    "message": "Transaction benefit already exists",  
    "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

5. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionBenefitType: +name

6. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{  
    "status": 409,  
    "message":  
        "Transaction amount must be greater than zero.",  
        "Transaction amount exceed the allowed limit of  
        9999999999999999.99.",  
        "Transaction amount must have at most two decimal places."  
    "errors": []  
}
```


GET TRANSACTION BENEFIT

Request

URI: /transaction-benefit/{id}

Method: GET

Response

1. Success - OK

Code 200 – OK

Body:

```
{  
  "id": UUID – "0000-ffff-0000-ffff",  
  "name": TransactionBenefitType – "ENUM",  
  "description": String – "string",  
  "amount": BigDecimal(18,2) – 00.00,  
  "date": LocalDate – "0000-00-00",  
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",  
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",  
  "user_id": UUID – "0000-ffff-0000-ffff"  
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid UUID Format",  
  "errors": []  
}
```

Request

URI: /transaction-income

Query Params: name, description, amount, date or no params to get all

Method: GET

Response:

1. Success

Code 200 – OK

Body:

```
{  
  "id": UUID – "0000-ffff-0000-ffff",  
  "name": TransactionIncomeType – "ENUM",  
  "description": String – "string",  
  "amount": BigDecimal(18,2) – 00.00,  
  "date": LocalDate – "0000-00-00",  
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",  
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",  
  "user_id": UUID – "0000-ffff-0000-ffff"  
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Name - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid value Provided for TransactionBenefitType: +name",  
  "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD

5. End date before Start date

Code: 400 – Bad Request

Body

```
{  
    "status": 400,  
    "message": "Start date cannot be after end date.",  
    "errors": []  
}
```

PUT TRANSACTION BENEFIT

Request

URI: /transaction-income/{id}

Method: PUT

Response

1. Success - OK

Code 204 – No Content

2. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

3. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionIncomeType: +name

4. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 409,  
  "message":  
    "Transaction amount must be greater than zero.",  
    "Transaction amount exceed the allowed limit of  
    9999999999999999.99.",  
    "Transaction amount must have at most two decimal places."  
  "errors": []  
}
```

DELETE TRANSACTION BENEFIT

Request

URI: /transaction-income/{id}

Method: DELETE

Response

1. Success - OK

Code 204 – No Content

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid UUID string: + id",  
  "errors": []  
}
```

POST TRANSACTION EXPENSE:

Request

URI: /transaction-expense

Method: POST

Body:

```
{  
    "name": TransactionExpenseType – "ENUM",  
    "description": String – "string",  
    "amount": BigDecimal(18,2) – 00.00,  
    "date": LocalDate – "0000-00-00",  
    "recurrent": Boolean – true  
}
```

Response

1. Success - OK

Code: 201 – Created

Header: Location – URI of created resource

2. Validation Error - OK

Code: 422 – Unprocessable Entity

Body:

```
{  
    "status": 422,  
    "message":  
        "Transaction expense name cannot be null",  
        "Transaction expense amount cannot be null",  
        "Transaction expense date cannot be null",  
        "Transaction expense recurrent cannot be null"  
    "errors": []  
}
```

3. Duplicated Transaction - OK

Code: 409 – Conflict

Body:

```
{  
    "status": 409,  
    "message": "Transaction expense already exists",  
    "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

5. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionExpenseType: +name

6. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{  
    "status": 409,  
    "message":  
        "Transaction amount must be greater than zero.",  
        "Transaction amount exceed the allowed limit of  
        9999999999999999.99.",  
        "Transaction amount must have at most two decimal places."  
    "errors": []  
}
```

GET TRANSACTION EXPENSE

Request

URI: /transaction- expense/{id}

Method: GET

Response

1. Success - OK

Code 200 – OK

Body:

```
{
  "id": UUID – "0000-ffff-0000-ffff",
  "name": TransactionExpenseType – "ENUM",
  "description": String – "string",
  "amount": BigDecimal(18,2) – 00.00,
  "date": LocalDate – "0000-00-00",
  "recurrent": Boolean - false
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",
  "user_id": UUID – "0000-ffff-0000-ffff"
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{
  "status": 400,
  "message": "Invalid UUID Format",
  "errors": []
}
```

Request

URI: /transaction- expense

Query Params: name, description, amount, date, recurrent or no params to get all

Method: GET

Response:

1. Success

Code 200 – OK

Body:

```
{  
  "id": UUID – "0000-ffff-0000-ffff",  
  "name": TransactionExpenseType – "ENUM",  
  "description": String – "string",  
  "amount": BigDecimal(18,2) – 00.00,  
  "date": LocalDate – "0000-00-00",  
  "recurrent": Boolean - false  
  "createdAt": LocalDateTime – "0000-00-00h00.00.00",  
  "updatedAt": LocalDateTime – "000-00-00h00.00.00",  
  "userId": UUID – "0000-ffff-0000-ffff"  
}
```

2. Error - OK

Code: 404 – Not Found

3. Invalid Name - OK

Code: 400 – Bad Request

Body:

```
{  
  "status": 400,  
  "message": "Invalid value Provided for TransactionExpenseType: +name",  
  "errors": []  
}
```

4. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD

5. End date before Start date

Code: 400 – Bad Request

Body

```
{  
    "status": 400,  
    "message": "Start date cannot be after end date.",  
    "errors": []  
}
```

PUT TRANSACTION EXPENSE

Request

URI: /transaction- expense/{id}

Method: PUT

Response

1. Success - OK

Code 204 – No Content

2. Invalid Date – OK

Code: 400 – Bad Request

Invalid date format. Expected format is YYYY-MM-DD.

3. Invalid Name – OK

Code: 400 – Bad Request

Invalid value provided for TransactionExpenseType: +name

4. Invalid Amount – OK

Code: 400 – Bad Request

Body:

```
{
  "status": 409,
  "message":
    "Transaction amount must be greater than zero.",
    "Transaction amount exceed the allowed limit of
    9999999999999999.99.",
    "Transaction amount must have at most two decimal places."
  "errors": []
}
```

DELETE TRANSACTION EXPENSE

Request

URI: /transaction- expense/{id}

Method: DELETE

Response

1. Success - OK

Code 204 – No Content

2. Error - OK

Code: 404 – Not Found

3. Invalid Format - OK

Code: 400 – Bad Request

Body:

```
{
  "status": 400,
  "message": "Invalid UUID string: + id",
  "errors": []
}
```

