

1 - Crie um endpoint “<http://localhost:8080/transaction/>” que receba uma request POST de um objeto Transaction, e armazene em um banco de dados conforme o exemplo abaixo.

POST /transaction HTTP/1.1

Host: localhost:8080

Content-Type: application/json

cache-control: no-cache

Payload:

```
{
  "date": "01/01/2019", // LocalDate
  "time": "13:52:33", // LocalDateTime
  "value": 100.00, // BigDecimal
  "cardApplication": "CREDITO", // enum CardApplication contendo os valores
  //DEBITO, CREDITO e VOUCHER
  "status": "SUCCESS" // enum PaymentStatus contendo os valores
  //SUCCESS, PENDING, CANCELED e FAILED
}
```

2 - Crie um endpoint “<http://localhost:8080/transaction/all/>” que receba uma request GET e liste todas as Transaction armazenadas no banco de dados.

GET /transaction/all HTTP/1.1

Host: localhost:8080

Content-Type: application/json

cache-control: no-cache

Response:

```
[
  {
    "id": 1,
    "date": "01/01/2019",
    "time": "13:52:33",
    "value": 100.00,
```

```
    "cardApplication": "CREDITO",
    "status": "SUCCESS"
  },
  {
    "id": 2,
    "date": "01/01/2019",
    "time": "13:52:33",
    "value": 100.00,
    "cardApplication": "CREDITO",
    "status": "SUCCESS"
  },
  ...
]
```

3 - Crie um endpoint “<http://localhost:8080/transaction/>” que receba uma request PUT com um objeto Transaction contendo o identificador e atualize o registro no banco de dados, conforme os dados recebidos no payload.

PUT /transaction/all HTTP/1.1

Host: localhost:8080

Content-Type: application/json

cache-control: no-cache

Payload:

```
{
  "id": 2,
  "date": "01/01/2019",
  "time": "13:52:33",
  "value": 100.00,
  "cardApplication": "CREDITO",
  "status": "CANCELED"
}
```