Credit Card Processing API- Access Guide

Version: 0.0.1-SNAPSHOT

Table of Contents

Overview

Credit Card Api is a RestFul application built with Spring Boot and Embedded MongoDB It's based on Level 3 of the famous [Richardson Maturity Model](#). This is one of the most discussed subjects about API design. This source code is hosted in https://github.com/rahulmzn/credit-card-processing.

Endpoints

Environment	Url
dev	http://localhost:8080/api
test	http://localhost/api
production	https://localhost/api

HTTP verbs

RESTful notes tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

Verb	Usage
GET	Used to retrieve a resource
POST	Used to create a new resource

HTTP status codes

RESTful notes tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP status codes.

Status code	Usage
200 OK	The request completed successfully
201 Created	A new resource has been created successfully. The resource's URI is available from the response's `Location` header
204 No Content	An update to an existing resource has been applied successfully

Status code	Usage
400 Bad Request	The request was malformed. The response body will include an error providing further information
422 Unprocessable Entity	The request was well-formed but was unable to be followed due to semantic errors.
404 Not Found	The requested resource did not exist

Hypermedia

RESTful Notes uses hypermedia and resources include links to other resources in their responses. Responses are in Hypertext Application Language (HAL) format. Links can be found benath the `_links` key. Users of the API should not created URIs themselves, instead they should use the above-described links to navigate from resource to resource.

Resources

Card

ⓘ │ The cards resource provides the entry point to add new card and get existing cards.

Accessing the card resource

version 1 (v1)

How to get existing cards

Request

```
GET /api/v1/cards HTTP/1.1
Content-Type: application/json
Accept-Encoding: gzip
Host: localhost:8080
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 462

[ {
  "id" : "1234",
  "name" : "Rahul Kumar",
  "number" : "4716651077977392",
  "limit" : "1000",
  "balance" : 10.0,
  "brand" : "VISA"
}, {
  "id" : "5ff291fa34e50353b6d0d66b",
  "name" : "Rahul Kumar",
  "number" : "4716651077977392",
  "limit" : "1000",
  "balance" : 10.0,
  "brand" : "VISA"
}, {
  "id" : "5ff291fb34e50353b6d0d66c",
  "name" : "Rahul Kumar",
  "number" : "4716651077977392",
  "limit" : "1000",
  "balance" : 10.0,
  "brand" : "VISA"
} ]
```

Response Fields

Path	Type	Description
[].id	String	Credit card unique ID.
[].name	String	Name of credit card Holder
[].number	String	Credit card Number
[].balance	Number	Available credit card Balance
[].limit	String	Total limit of credit card
[].brand	String	Brand of credit card ex. VISA

Add credit card

> When the request is completed successfully, the service return a 201 http status code,
> If given inputs are not valid then service will return 422 http status .

Request Fields

Path	Type	Description
id	String	Credit card unique ID.
name	String	Name of credit card Holder
number	String	Credit card Number
limit	String	Total limit of credit card
brand	String	Brand of credit card ex. VISA
balance	Number	Available credit card Balance

Request

```
POST /api/v1/cards HTTP/1.1
Content-Type: application/json
Accept-Encoding: gzip
Content-Length: 138
Host: localhost:8080

{
  "id" : "1234",
  "name" : "Rahul Kumar",
  "number" : "4716651077977392",
  "limit" : "1000",
  "balance" : 10.0,
  "brand" : "VISA"
}
```

Response

```
HTTP/1.1 201 Created
```

Card data validation exception. This occurs with invalid card parameters

Request

```
POST /api/v1/cards HTTP/1.1
Content-Type: application/json
Accept-Encoding: gzip
Content-Length: 108
Host: localhost:8080

{
  "id" : null,
  "name" : null,
  "number" : null,
  "limit" : null,
  "balance" : 0.0,
  "brand" : null
}
```

Response

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Content-Length: 597

{
  "code" : "176cb8a3dc5",
  "exception" : "InvalidRequestException",
  "message" : "Invalid Card",
  "errors" : [ {
    "code" : "NotNull",
    "message" : "must not be null",
    "parameter" : "number"
  }, {
    "code" : "NotNull",
    "message" : "must not be null",
    "parameter" : "brand"
  }, {
    "code" : "NotNull",
    "message" : "must not be null",
    "parameter" : "name"
  }, {
    "code" : "NotNull",
    "message" : "must not be null",
    "parameter" : "limit"
  }, {
    "code" : "CardNumber",
    "message" : "invalid credit card number",
    "parameter" : "number"
  } ]
}
```