Documentation for Payment Gateway API

Tested on : sts-4.8.1.RELEASE; JAVA 11

To work with this project, import in STS as Existing Maven Project and right click on project name Project Name -> Run As -> Spring Boot App.

Following Headers need to be sent with every request in the headers:

Content-Type : application/json

X-UID : 4ed2319b72e66b091ad2129

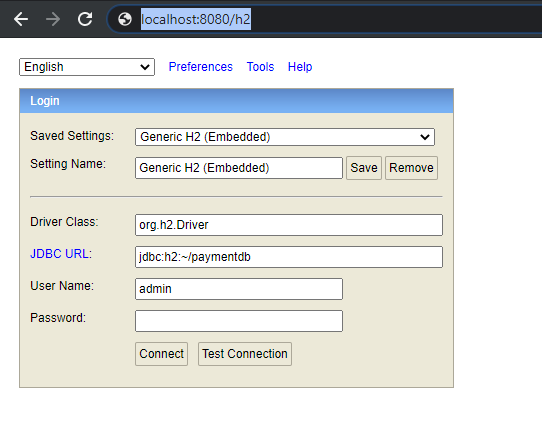


X-UID is here serving as an authToken to verify the request is coming from an authentic resource. This is validated before each request. If User is not valid; status will be set to "Unauthorized User" in the returned json and status 403(FORBIDDEN) will be returned.

Database:

An in-memory SQL Database H2 has been used to store details about transactions. As soon as project is booted on localhost you can access

<http://localhost:8080/h2> to access this DB and check details inside it.



Details about this (user/pass/connection url) are specified in the application.properties file.

Validations:

Validations have been put on some of the fields to produce custom Error messages in case some Unexpected Input is given.

For Example, if Card type is anything other than “debitcard” or “creditcard”; we’ll get an error message saying **"Card type must be debit or credit"** specifying also the json field this error occurred on.

Defined APIs:

1) Endpoint : “/transaction”

Request Type : “POST”

Parameters : amount, currency, type, card (number,expirationMonth, expirationYear,cvv)

Headers : X-UID, Content-Type

Description : Used to save transaction details to a Database

Example of a Successful Response: (Status 200)

Request sent in json raw format from Postman:

{

"amount": "100",

"currency": "USD",

"type": "creditcard",

"card": {

"number": "4111111111111111",

"expirationMonth": "2",

"expirationYear": "2020",

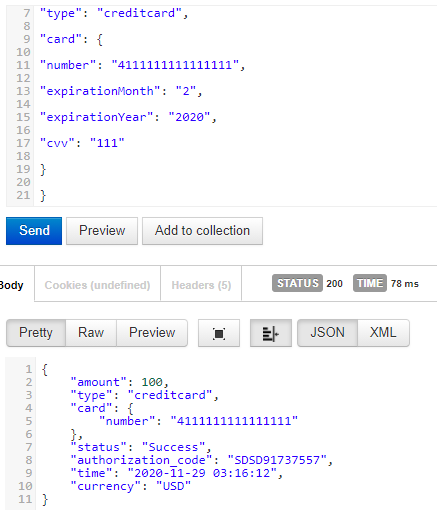
"cvv": "111"

}

}

Response:

{ "amount": 100, "type": "creditcard", "card": { "number": "4111111111111111" }, "status": "Success", "authorization\_code": "SDSD58562828", "time": "2020-11-29 02:33:41", "currency": "USD"}



Example of Error Response: (Status 400)

Request sent in json raw format from Postman:

{

"amount": "100",

"currency": "USD",

"type": "mycard"

}

Response:

[

{

"fieldName": "card",

"message": "Card Details cannnot be null"

},

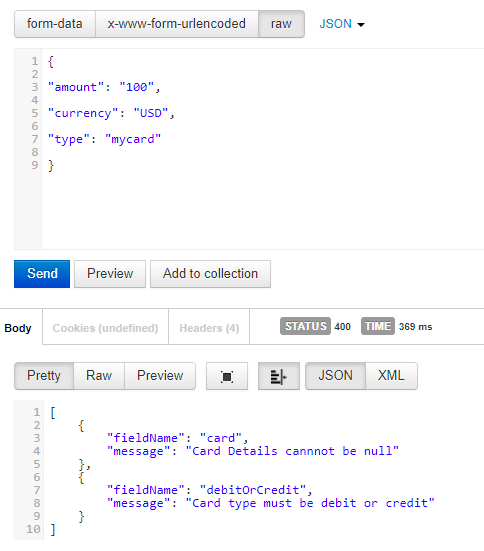
{

"fieldName": "debitOrCredit",

"message": "Card type must be debit or credit"

}

]



2) Endpoint : “/transaction”

Request Type : “GET”

Parameters : None

Headers : X-UID, Content-Type

Description : Used to get all Current Transactions saved in the Database

Example of a Successful Response: (Status 200)

Request sent in json raw format from Postman:

Get request with valid headers to “/transaction”

Response:

[

{

"amount": 100,

"type": "creditcard",

"card": {

"number": "4111111111111111"

},

"status": "Success",

"authorization\_code": "SDSD16042270",

"time": "02:54:08",

"currency": "USD"

},

{

"amount": 500,

"type": "debitcard",

"card": {

"number": "10934809018"

},

"status": "Success",

"authorization\_code": "SDSD98389162",

"time": "02:54:44",

"currency": "CAD"

},

{

"amount": 1500000,

"type": "debitcard",

"card": {

"number": "2908409813123"

},

"status": "Success",

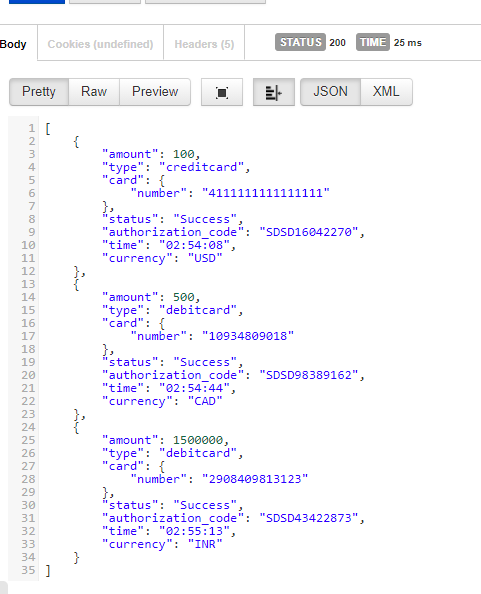
"authorization\_code": "SDSD43422873",

"time": "02:55:13",

"currency": "INR"

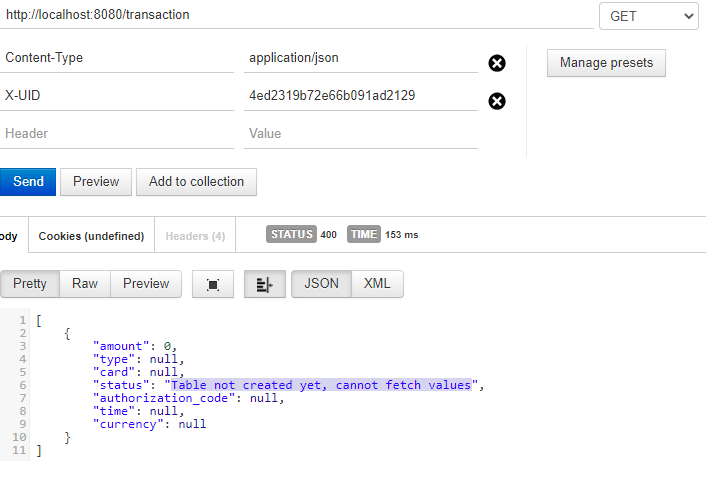
}

]



Note:

If a get request is made without even making a single post request; Status 400 BAD REQUEST will be returned with response as follows:



as we are trying to query on a table that has not been created till now.

Assumptions:

Following things were taken as assumptions:

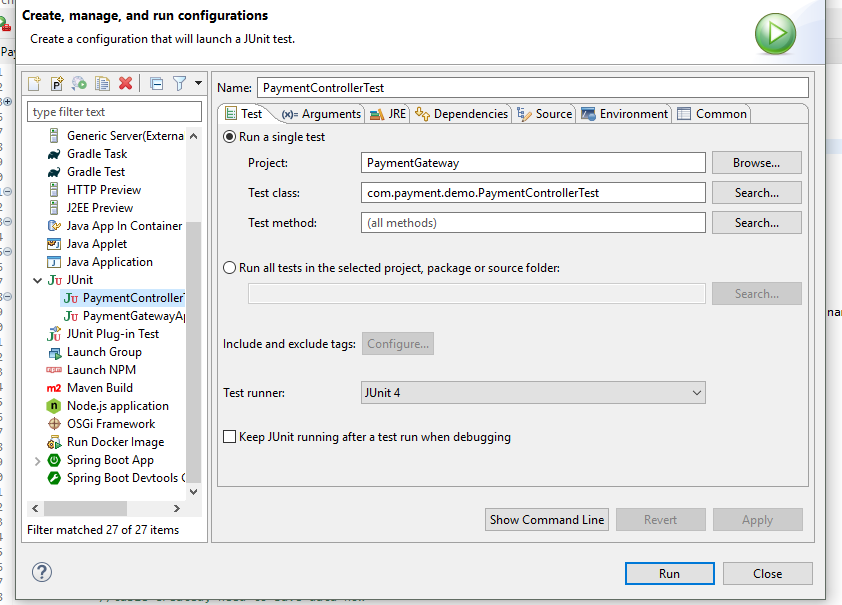
1) Client will be provided with X-UID to provide in headers when making request. This was done for security purpose.

2) authorization\_code field has been assumed to have the prefix “SDSD” followed by a random 8 digit numeric value. This is generated uniquely for every request.

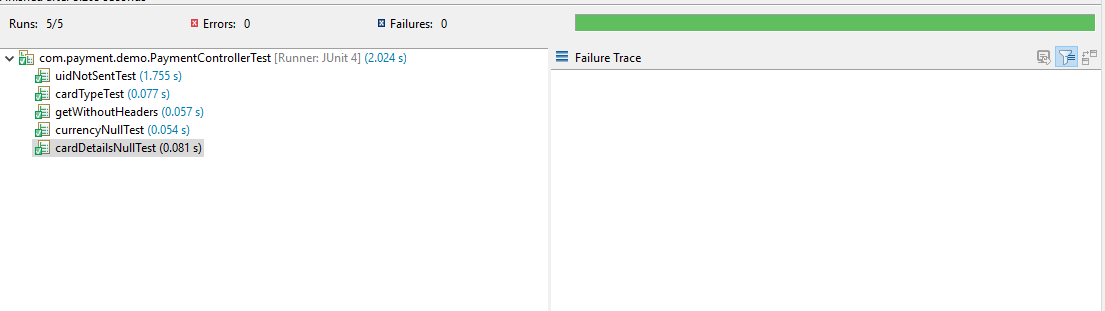
3) jdbc is used to connect and run queries in H2 database.

Testing:

To run tests make sure that Junit version is set to 4. For this right click project name. Go to Run as -> Run configurations -> Junit (on left tab; expand it) -> PaymentControllerTest and check that version is set to Junit4



To run any test just right click on PaymentControllerTest.java inside src/test/java and run as “Junit Test”. Following is the result:



-Rythm Sharma