Payment REST API Design Document

Methods

1. Create a Payment

Allows for a Payment to be received and persisted to the database.
The payment will be given a generated id.
Note: this method is not idempotent, if you post the same payment JSON data twice, then two payments will be created with different generated ids.

Request

METHOD	POST
URL	/payments
REQUIRED CONTENT	JSON representation of a Payment object

EXAMPLE REQUEST BODY	{
	 "type": "Payment",
	 "version": 6,
	 "organisation_id": "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
	 "attributes": {
	  "amount": "100.21",
	  "beneficiary_party": {
	   "account_name": "W Owens",
	   "account_number": "31926819",
	   "account_number_code": "BBAN",
	   "account_type": 0,
	   "address": "1 The Beneficiary Localtown SE2",
	   "bank_id": "403000",
	   "bank_id_code": "GBDSC",
	   "name": "Wilfred Jeremiah Owens"
	  },
	  "charges_information": {
	   "bearer_code": "SHAR",
	   "sender_charges": [
	    {
	     "amount": "5.00",
	     "currency": "GBP"

```
        },
        {
          "amount": "10.00",
          "currency": "USD"
        }
      ],
      "receiver_charges_amount": "1.00",
      "receiver_charges_currency": "USD"
    },
    "currency": "GBP",
    "debtor_party": {
      "account_name": "EJ Brown Black",
      "account_number": "GB29XABC10161234567801",
      "account_number_code": "IBAN",
      "address": "10 Debtor Crescent Sourcetown NE1",
      "bank_id": "203301",
      "bank_id_code": "GBDSC",
      "name": "Emelia Jane Brown"
    },
    "end_to_end_reference": "Wil piano Jan",
    "fx": {
      "contract_reference": "FX123",
      "exchange_rate": "2.00000",
      "original_amount": "200.42",
      "original_currency": "USD"
    },
    "numeric_reference": "1002001",
    "payment_id": "123456789012345678",
    "payment_purpose": "Paying for goods/services",
    "payment_scheme": "FPS",
    "payment_type": "Credit",
    "processing_date": "2017-01-18",
    "reference": "Payment for Em's piano lessons",
    "scheme_payment_sub_type": "InternetBanking",
    "scheme_payment_type": "ImmediatePayment",
    "sponsor_party": {
      "account_number": "56781234",
      "bank_id": "123123",
      "bank_id_code": "GBDSC"
    }
  }
}
```

Success Response

HTTP STATUS CODE	201 Created
HEADERS	Location : url of newly created resource, will be of format: /payments/{generated id}

Error Response

HTTP STATUS CODE	**REASON**
400 Bad Request	If the posted body contained invalid content and could not be parsed into a valid Payment object.

2. Fetch a Payment

Allows for a Payment to be read from the database.

Request

METHOD	GET
URL	/payments/{id}

Success Response

HTTP STATUS CODE	200 Ok

Error Response

HTTP STATUS CODE	**REASON**
404 Not found	If the id requested does not match an existing payment

3. Fetch a Collection of Payments

Allows for a collection of Payments to be read from the database. Should support paging and sorting of results.

Request

METHOD	GET
URL	/payments
OPTIONAL QUERY PARAMETERS	page : to return a given page size : to specify a page size sort : to specify a sort criteria for the data.

Success Response

HTTP STATUS CODE	200 ok

4. Update a Payments

Allows for a given payment to be overwritten with new data by specifying its Id

Request

METHOD	PUT
URL	/payments/{id}
REQUIRED	JSON representation of a Payment object

CONTENT	

Success Response

HTTP STATUS CODE	200 ok

Error Response

HTTP STATUS CODE	**REASON**
400 bad request	If the put body contained invalid content and could not be parsed into a valid Payment object.

5. Partially Update a Payment

Allows for a given payment to be partially changed by sending a JSON representation of a sub-set of its data.

Request

METHOD	PATCH
URL	/payments/{id}
REQUIRED CONTENT	Partial JSON representation of a Payment object, e.g. {"organisation_id":"newOrganisationId"} To just update the organisation_id field

Success Response

HTTP	200 ok

STATUS CODE	

Error Response

HTTP STATUS CODE	**REASON**
400 bad request	If the body contained invalid content and could not be parsed into a valid partial representation of a Payment object.
404 Not found	If the id in the request url does not match an existing payment

6. Delete a Payment

Allows for a given payment to be deleted from the database.

Request

METHOD	DELETE
URL	/payments/{id}

Success Response

HTTP STATUS CODE	200 ok

Error Response

HTTP STATUS CODE	**REASON**
404 Not found	If the id in the request url does not match an existing payment

Technical Design

The API should support the principal of HATEOAS (Hypermedia as the Engine of Application State).
To do this it should return JSON representations of type: Hypertext Application Language (HAL).
The API should be built in Java using ‘spring-data-rest’ to minimise the amount of code written and the need for testing (no need to test extensively the built in default functionality of the spring-data-rest framework).