## Design document for the “Coding Exercise” application

The application exposes a set of RESTful endpoints that allows manipulation of payments. The application behaviour is described bellow:

- To retrieve the collection of payment resources:

GET /v1/payments
Content-Type: application/json;charset=UTF-8

Response status: 200
Response body:

```
{
  "data": [ {...} ],
  "links": {
        "self": "https://api.test.form3.tech/v1/payments"
  }
}
```

Headers:

*content-type: application/json*
*content-length: 64*
*date: Fri, 08 Dec 2017 14:55:17 GMT*

The ‘data’ field is going to contain the list of payments, if there are any.

- To fetch a single payment resource by its identifier (UUID path parameter):

GET /v1/payments/{UUID}
Content-Type: application/json;charset=UTF-8

Will return the same API response as above with a single PaymentResource in the 'data' field, if there’s a payment registered with this identifier. Otherwise, an error is going to be returned:

Response status: 404
Response body:

```
{
  "status": 404,
  "message": "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb"
}
```

*The ‘message’ field contains the request identifier which was not found

If an invalid identifier is provided the response is going to be:

Response code: 400
Response body:

```
{
  "status": 400,
  "message": "Invalid UUID string: 743d5b"
}
```

- To create a payment resource do:

POST /v1/payments

With the request body of the payment you want to save.

If the request body is not provided the response is going to be:

Response code: 400
Response body:

```
{
  "status": 400,
  "message": "Missing mandatory argument"
}
```

The success response is going be:

Response code: 201

Response body is going to be an API response object with a single payment object in its ‘data’ field and the ‘links’ field is going to point at the newly created resource:

```
 "links": {
    "self": "http://localhost:8080/v1/payments/{UUID}"
  }
```

*UUID parameter is going to be generated by the application and will have a similar structure to “9568a24e-44c7-4b85-977e-4694fdc6da10”

If the request body is missing one or more mandatory fields a response with the appropriate message is going to be returned:

Response code: 400
Response body:

*{*
*"status": 400,*
*"message": "NULL not allowed for column \"TYPE\"; SQL statement:\ninsert into payment_resource_entity … /* error related information */*
*}*

- To update a payment resource:

PUT /v1/payments/{UUID}

With the request body of the payment you want to overwrite the existing one with.

If the requested UUID is not found, malformed or the body is missing a mandatory field the behaviour is going to be the same as for the methods described above.

If the request is accepted the response will be:

Response code: 204
Response body: emtpy

- To delete a know resource:

DELETE /v1/payments/{UUID}

If the requested identifier is not found or is malformed then the behaviour is going to be the same as for the cases described previously.

If the server satisfied the request the response is going to be:

Response code: 204
Response body: emtpy

The behaviour of the RESTful interface conforms to the specifications described here: https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html