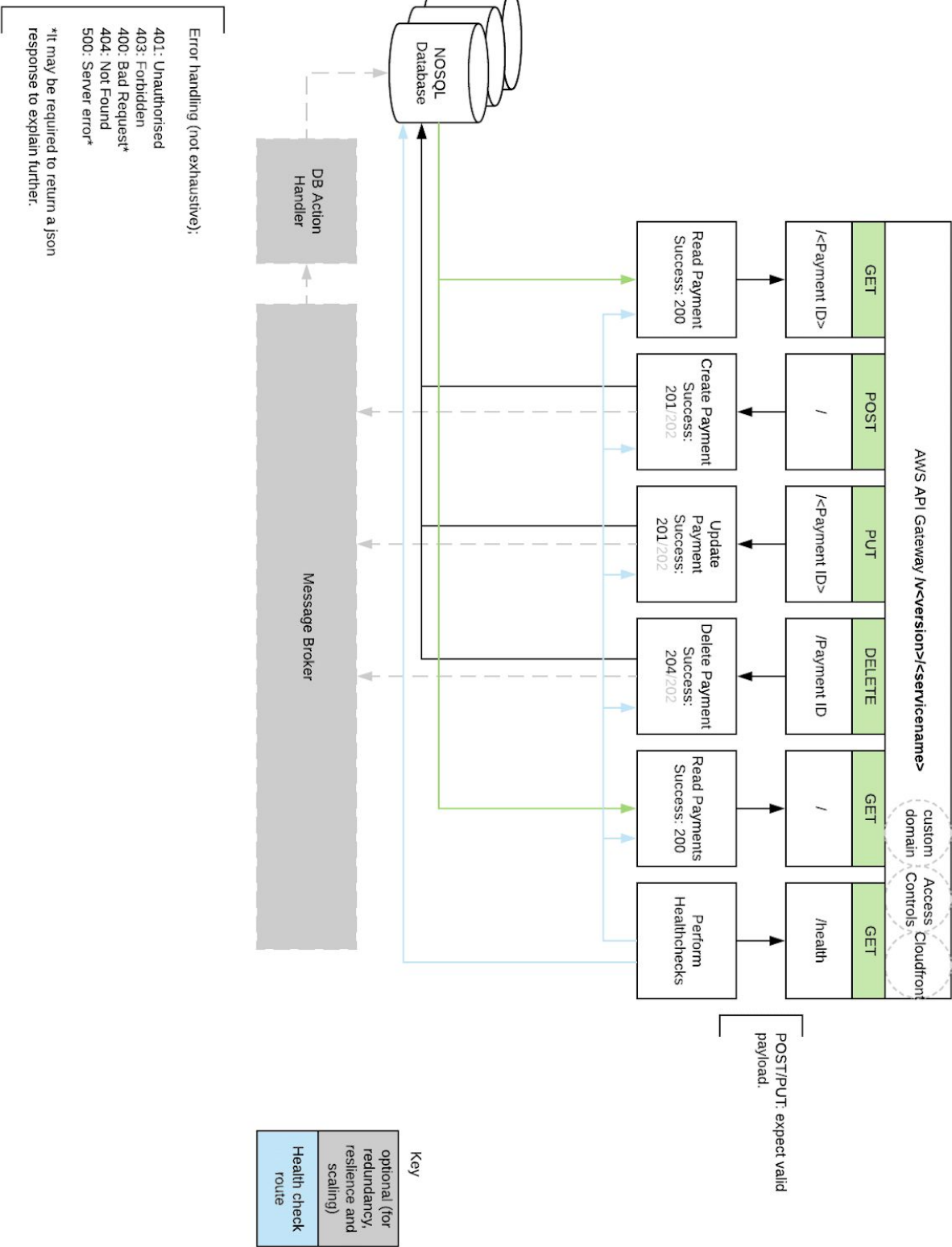


Payments API Design



## Notes:

- The design to be built is with a database and no message broker. But it's likely that a broker will be needed at some point.

## Data Structure:

See <https://github.com/cheekymonkat/payments/tree/master/src/test/resources/fixtures> for examples on data structure..

## GET /v1/payments/<ID>

Returns the data model

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

## GET /v1/payments/

Returns the data model

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

## POST /v1/payments/

Requires model:

```
{  
  ..payment..  
}
```

*Note: payment.id is not required and will be ignored.*

## PUT /v1/payments/<ID>

Requires model:

```
{  
  ..payment..  
}
```

*Note: payment.id must match that of <ID>*

## DELETE /v1/payments/<ID>

Returns: Status 204 if successful.

## Error handling

All errors return a response code with a json message in the format:

```
{  
  "message": "..error message..  
}
```