

Payments

Table of Contents

Overview.....	2
Version information	2
Contact information	2
License information	2
URI scheme	2
Consumes	2
Produces.....	3
External Docs	3
Paths	3
List all payments.....	3
Description	3
Responses	3
Bulk update payments.....	3
Description	3
Parameters	3
Responses	4
Create new payment.....	4
Description	4
Parameters	4
Responses	4
Example HTTP request.....	5
Example HTTP response	5
Delete all payments.....	5
Description	5
Responses	5
Find payment by ID.....	6
Description	6
Parameters	6
Responses	6
Example HTTP response	6
Update payment.....	7
Description	7
Parameters	7
Responses	7
Example HTTP request.....	7
Example HTTP response	8

Delete a payment.....	8
Description	8
Parameters	8
Responses	8
Definitions	9
Payment	9
PaymentAttributes.....	9
Party	11
Money.....	11
Charges.....	12
CurrencyExchange.....	12

Overview

This is a payments API design. The API will utilise the [go-json-rest](#) web framework backed by [gorm](#), connecting to a [postgres](#) database.

Version information

Version : 3.7.3

Contact information

Contact : Liam Humphreys

License information

License : MIT

License URL : <https://raw.githubusercontent.com/Angry-Potato/go-pay-me/master/LICENSE>

Terms of service : null

URI scheme

Host : localhost

BasePath : /

Schemes : HTTP

Consumes

- [application/json](#)

Produces

- `application/json`

External Docs

Description : Raw file on Github

URL : <https://raw.githubusercontent.com/Angry-Potato/go-pay-me/master/design/payments.swagger.json>

Paths

List all payments.

GET /payments

Description

Lists all payments.

Responses

HTTP Code	Description	Schema
200	Success	< Payment > array
500	Internal Server Error	No Content

Bulk update payments.

PUT /payments

Description

Bulk updates payments, as well as all their sub-resources.

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Payments to be updated.	< Payment > array

Responses

HTTP Code	Description	Schema
200	Success	< Payment > array
400	Bad Request	No Content
500	Internal Server Error	No Content

Create new payment.

POST /payments

Description

Creates a new payment, and all of its' sub-resources.

Parameters

Type	Name	Description	Schema
Body	body <i>required</i>	Payment that needs to be added to the service	Payment

Responses

HTTP Code	Description	Schema
201	Created	Payment
400	Bad Request	No Content
500	Internal Server Error	No Content

Example HTTP request

Request body

```
{
  "type" : "Payment",
  "id" : "216d4da9-e59a-4cc6-8df3-3da6e7580b77",
  "version" : 0,
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "attributes" : {
    "$ref" : "#/definitions/PaymentAttributes/example"
  }
}
```

Example HTTP response

Response 201

```
{
  "type" : "Payment",
  "id" : "216d4da9-e59a-4cc6-8df3-3da6e7580b77",
  "version" : 0,
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "attributes" : {
    "$ref" : "#/definitions/PaymentAttributes/example"
  }
}
```

Delete all payments.

DELETE /payments

Description

Deletes all payments, and their sub-resources.

Responses

HTTP Code	Description	Schema
200	Success	No Content
500	Internal Server Error	No Content

Find payment by ID.

GET /payments/{paymentId}

Description

Returns a single payment

Parameters

Type	Name	Description	Schema
Path	paymentId <i>required</i>	ID of payment to return	string (uuid)

Responses

HTTP Code	Description	Schema
200	Success	Payment
400	Bad Request	No Content
404	Not found	No Content
500	Internal Server Error	No Content

Example HTTP response

Response 200

```
{
  "type" : "Payment",
  "id" : "216d4da9-e59a-4cc6-8df3-3da6e7580b77",
  "version" : 0,
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "attributes" : {
    "$ref" : "#/definitions/PaymentAttributes/example"
  }
}
```

Update payment.

PUT /payments/{paymentId}

Description

Updates a payment with new information.

Parameters

Type	Name	Description	Schema
Path	paymentId <i>required</i>	ID of payment to update	string (uuid)
Body	body <i>required</i>	Updated payment	Payment

Responses

HTTP Code	Description	Schema
200	Success	Payment
304	Not Modified	No Content
400	Bad Request	No Content
404	Not found	No Content
500	Internal Server Error	No Content

Example HTTP request

Request body

```
{
  "type" : "Payment",
  "id" : "216d4da9-e59a-4cc6-8df3-3da6e7580b77",
  "version" : 0,
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "attributes" : {
    "$ref" : "#/definitions/PaymentAttributes/example"
  }
}
```

Example HTTP response

Response 200

```
{
  "type" : "Payment",
  "id" : "216d4da9-e59a-4cc6-8df3-3da6e7580b77",
  "version" : 0,
  "organisation_id" : "743d5b63-8e6f-432e-a8fa-c5d8d2ee5fcb",
  "attributes" : {
    "$ref" : "#/definitions/PaymentAttributes/example"
  }
}
```

Delete a payment.

```
DELETE /payments/{paymentId}
```

Description

Deletes a specific payment, and its' sub-resources.

Parameters

Type	Name	Description	Schema
Path	paymentId <i>required</i>	ID of payment to delete	string (uuid)

Responses

HTTP Code	Description	Schema
200	Success	No Content
400	Bad Request	No Content
404	Not found	No Content
500	Internal Server Error	No Content

Definitions

Payment

A payment between two parties.

Name	Description	Schema
type <i>optional</i>	Default : "Payment"	enum (Payment)
id <i>required</i>		string (uuid)
version <i>optional</i>	Default : 0	integer (int32)
organisation_id <i>required</i>		string (uuid)
attributes <i>required</i>		PaymentAttributes

PaymentAttributes

Details of a payment.

Name	Schema
amount <i>required</i>	string (money)

Name	Schema
beneficiary_party <i>required</i>	Party
charges_information <i>required</i>	Charges
currency <i>required</i>	string
debtor_party <i>required</i>	Party
end_to_end_reference <i>required</i>	string
fx <i>optional</i>	CurrencyExchange
numeric_reference <i>required</i>	string
payment_id <i>required</i>	string
payment_purpose <i>required</i>	string
payment_scheme <i>required</i>	string
payment_type <i>required</i>	enum (Credit)
processing_date <i>required</i>	string
reference <i>required</i>	string
scheme_payment_sub_type <i>required</i>	enum (InternetBanking)

Name	Schema
scheme_payment_type <i>required</i>	enum (ImmediatePayment)
sponsor_party <i>required</i>	Party

Party

A real-life entity capable of making and receiving payments.

Name	Schema
account_name <i>optional</i>	string
account_number <i>required</i>	string
account_number_code <i>optional</i>	string
account_type <i>optional</i>	integer (int32)
address <i>optional</i>	string
bank_id <i>required</i>	string
bank_id_code <i>required</i>	string
name <i>optional</i>	string

Money

Makes the world go round.

Name	Schema
amount <i>required</i>	string (money)
currency <i>required</i>	string

Charges

A list of monies to be deducted from somewhere.

Name	Schema
bearer_code <i>required</i>	string
sender_charges <i>required</i>	< Money > array
receiver_charges_amount <i>required</i>	string (money)
receiver_charges_currency <i>required</i>	string

CurrencyExchange

An amount being exchanged from a given currency.

Name	Schema
contract_reference <i>required</i>	string
exchange_rate <i>required</i>	string
original_amount <i>required</i>	string (money)
original_currency <i>required</i>	string