

Payment API

F3 Payment API <https://github.com/brunotm/f3api> (<https://github.com/brunotm/f3api>)

Version 0.0.1

Contact information

brunotm@gmail.com (<mailto:brunotm@gmail.com>)

License

Apache 2.0 (<http://www.apache.org/licenses/LICENSE-2.0.html>)

Paths

`/payment`

POST /payment

payment

Summary

Create a payment

Parameters

Name	Located in	Description	Required	Schema
body	body	Payment object	Yes	<pre>↔ ▼Payment { id: string organisation_id: string type: string version: integer attributes: ▶Attributes { } }</pre>

Responses

Code	Description	Schema
200	payment created	<pre>↔ ▼Response { reason: string links: ▶links { } }</pre>
400	invalid payment	<pre>↔ ▼Response { reason: string links: ▶links { } }</pre>

Try this operation

/payment/{paymentId}

GET /payment/{paymentId}

payment

Summary

Get a payment

Parameters

Name	Located in	Description	Required	Schema
paymentId	path	Payment ID	Yes	⇔ string

Responses

Code	Description	Schema
200	payment found	<pre>▼Payment { id: string organisation_id: string type: string version: integer attributes: ▶Attributes { } }</pre>
400	invalid payment id	<pre>▼Response { reason: string links: ▶links { } }</pre>
404	payment not found	<pre>▼Response { reason: string links: ▶links { } }</pre>

Try this operation

PUT /payment/{paymentId}

payment

Summary

Update a payment

Parameters

Name	Located in	Description	Required	Schema
paymentId	path	Payment ID to be updated	Yes	⇔ string
body	body	Payment object	Yes	⇔ ▼ Payment { id: string organisation_id: string type: string version: integer attributes: ►Attributes { } }

Responses

Code	Description	Schema
200	payment updated	⇔ ▼ Response { reason: string links: ►links { } }
400	invalid payment id	⇔ ▼ Response { reason: string links: ►links { } }
404	payment not found	⇔ ▼ Response { reason: string links: ►links { } }

Try this operation

DELETE /payment/{paymentId}

payment

Summary

Delete a payment

Parameters

Name	Located in	Description	Required	Schema
paymentId	path	Payment ID to delete	Yes	⇒ string

Responses

Code	Description	Schema
200	successful operation	⇒ ▼Response { reason: string links: ►links { } }
400	invalid payment Id	⇒ ▼Response { reason: string links: ►links { } }
404	payment not found	⇒ ▼Response { reason: string links: ►links { } }

Try this operation

/payments

GET /payments

payments

Summary

Get all payments

Responses

Code	Description
200	successful operation

Schema

```
▼ Payments {  
  data: ▶ []  
  links: ▶ links { }  
}
```

Try this operation

Models

Payment

```
▼ Payment {  
  id: string  
  organisation_id: string  
  ⇔ type: string  
  version: integer  
  attributes: ▶ Attributes { }  
}
```

Attributes

```

▼Attributes {
  amount:          string
  currency:        string
  end_to_end_reference: string
  numeric_reference: string
  payment_id:      string
  payment_purpose:   string
  payment_scheme:  string
  payment_type:    string
⇒ processing_date: string
  reference:       string
  scheme_payment_type: string
  scheme_payment_sub_type: string
  fx:              ►FX { }
  sponsor_party:   ►Party { }
  beneficiary_party: ►Party { }
  debtor_party:    ►Party { }
  charges_information: ►Charges[]
}

```

FX

```

▼FX {
  contract_reference: string
⇒ exchange_rate:      string
  original_amount:    string
  original_currency:  string
}

```

Party

```
▼Party {
  name: string
  account_name: string
  account_number: string
⇒ account_number_code: string
  account_type: integer
  address: string
  bank_id: string
  bank_id_code: string
}
```

Charge

```
▼Charge {
⇒ amount: string
  currency: string
}
```

Charges

```
▼Charges [
⇒ ►Charge { }
]
```

Payments

```
▼Payments {
⇒ data: ►[]
  links: ►links { }
}
```

Response

```
▼Response {
⇒ reason: string
}
```



```
links: ▶links { }  
}
```

links

```
▼links {  
⇒ self: string  
}
```