

# Payment API

Demo code

**Version** 1.0.0

**License**

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

## Paths


/payments

GET /payments

### Summary

Lists payments

### Responses

Code	Description	Schema
200	The operation was successful	 <pre>▼ [   ▶ PaymentList { }</pre>
500	There was a problem inside the server	

Try this operation

## POST /payments

### Summary

Creates a new payment

### Parameters

Name	Located in	Description	Required	Schema
body	body	Payment object to be added	Yes	<div><div>↔</div><div><div>▼ Payment {</div><div>type: string</div><div>id: string</div><div>version: number</div><div>organisation_id: string</div><div>attributes: ▶ Payment_attributes { }</div></div></div>

### Responses

Code	Description
200	Payment successfully added
400	Returned due to invalid input, including when an ID for the payment is not provided
500	There was a problem inside the server

Try this operation

/payments/{payment\_id}

GET /payments/{payment\_id}

## Summary

Gets a payment

## Parameters

Name	Located in	Description	Required	Schema
payment_id	path	ID of payment	Yes	↔ string (uuid)

## Responses

Code	Description	Schema
<b>200</b>	The operation was successful	↔ ▼ { }
<b>404</b>	Payment not found	
<b>500</b>	There was a problem inside the server	

Try this operation

PUT /payments/{payment\_id}

## Summary

Update a payment

## Parameters

Name	Located in	Description	Required	Schema
payment_id	path	ID of payment	Yes	⇔ string (uuid)
body	body	Payment object	Yes	⇔ <div>▼ <b>Payment</b> {   type: string   id: string   version: number   organisation_id: string   attributes: ► <b>Payment_attributes</b> { } }</div>

## Responses

Code	Description
<b>200</b>	The operation was successful
<b>400</b>	Invalid content or Invalid ID supplied
<b>404</b>	Payment not found
<b>500</b>	There was a problem inside the server

Try this operation

DELETE /payments/{payment\_id}

## Summary

Deletes a payment

## Parameters

Name	Located in	Description	Required	Schema
payment_id	path	ID of payment	Yes	↔ string (uuid)

## Responses

Code	Description
<b>200</b>	The operation was successful
<b>404</b>	Payment not found
<b>500</b>	There was a problem inside the server

Try this operation

## Models

### PaymentList

```
▼ PaymentList {  
  ↔ data: ▶ []  
}
```

### Payment

```
▼ Payment {
```

```
    type:          string
    id:            string
    ↺ version:      number
    organisation_id: string
    attributes:    ▶ Payment_attributes { }
  }
```

## Payment\_attributes\_beneficiary\_party

```
  ▼ Payment_attributes_beneficiary_party {
    account_name:      string
    account_number:    string
    account_number_code: string
    ↺ account_type:     number
    address:           string
    bank_id:           string
    bank_id_code:      string
    name:              string
  }
```

## Payment\_attributes\_charges\_information\_sender\_charges

```
  ▼ Payment_attributes_charges_information_sender_charges {
    ↺ amount:  string
    currency: string
  }
```

## Payment\_attributes\_charges\_information

```
  ▼ Payment_attributes_charges_information {
    bearer_code:      string
    ↺ sender_charges:  ▶ []
    receiver_charges_amount: string
  }
```

```
    receiver_charges_currency: string
}
```

## Payment\_attributes\_debtor\_party

```
▼ Payment_attributes_debtor_party {
  account_name:      string
  account_number:    string
  account_number_code: string
⇒ address:           string
  bank_id:           string
  bank_id_code:      string
  name:              string
}
```

## Payment\_attributes\_fx

```
▼ Payment_attributes_fx {
  contract_reference: string
⇒ exchange_rate:      string
  original_amount:    string
  original_currency:  string
}
```

## Payment\_attributes\_sponsor\_party

```
▼ Payment_attributes_sponsor_party {
  account_number: string
⇒ bank_id:         string
  bank_id_code:   string
}
```

## Payment\_attributes

```

▼Payment_attributes {
  amount:                string
  beneficiary_party:     ►Payment_attributes_beneficiary_party { }
  charges_information:   ►Payment_attributes_charges_information { }
  currency:              string
  debtor_party:          ►Payment_attributes_debtor_party { }
  end_to_end_reference:  string
  fx:                    ►Payment_attributes_fx { }
  numeric_reference:     string
  payment_id:            string
  payment_purpose:         string
  payment_scheme:        string
  payment_type:          string
  processing_date:       string
  reference:             string
  scheme_payment_sub_type: string
  scheme_payment_type:   string
  sponsor_party:        ►Payment_attributes_sponsor_party { }
}

```

Try a sample here

# API Design First Workflow

Easily create API specifications