

# Swagger Example API

This is a sample server celler server.

**Version** 1.0

**Contact information**

**Terms of service**

<http://swagger.io/terms/> (<http://swagger.io/terms/>)

**License**

## Paths

/payments

## GET /payments








### Summary

Get the last 100 payments

### Description

Gets the last 100 payments. It should have a pagination

### Responses

Code	Description	Schema
<b>200</b>	OK	   <b>model.Payment { }</b>
<b>404</b>	Can not find ID	  { }
<b>500</b>	Cannot process the request	  { }

Try this operation

## POST /payments

### Summary

Create a new Payment

### Parameters

Name	Located in	Description	Required	Schema
payment	body	The payment to be created	Yes	<div><div>↔</div><div><div>▼model.Payment {</div><div>attributes: ▶model.Attributes { }</div><div>id: string</div><div>organisation_id: string</div><div>type: string</div><div>version: integer</div><div>}</div></div></div>

### Responses

Code	Description	Schema
204	Empty result, it is not the created object" TODO fix this	<div><div>↔</div><div><div>▼model.Payment {</div><div>attributes: ▶model.Attributes { }</div><div>id: string</div><div>organisation_id: string</div><div>type: string</div><div>version: integer</div><div>}</div></div></div>
400	Payment with invalid format	<div><div>↔</div><div><div>▼ {</div><div>}</div></div></div>
500	Cannot process the request	<div><div>↔</div><div><div>▼ {</div><div>}</div></div></div>

Try this operation

/payments/{paymentID}

GET /payments/{paymentID}

### Summary

Get a Payment by ID

### Parameters

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

### Responses

Code	Description	Schema
200	OK	<pre>▼model.Payment {   attributes:      ►model.Attributes { }   id:              string   organisation_id: string   type:            string   version:         integer }</pre>
404	Can not find ID	<pre>⇔ ▼ {   }</pre>
500	Cannot process the request	<pre>⇔ ▼ {   }</pre>

Try this operation

DELETE /payments/{paymentID}

## Summary

Delete a Payment by ID

## Parameters

Name	Located in	Description	Required	Schema
paymentID	path	Payment ID	Yes	$\Rightarrow$ string

## Responses

Code	Description	Schema
204	Empty result, it is not the created object" TODO fix this	$\Rightarrow$ <pre>▼model.Payment {   attributes:      ►model.Attributes { }   id:              string   organisation_id: string   type:            string   version:         integer }</pre>
404	Can not find ID	$\Rightarrow$ <pre>▼ { }</pre>
500	Cannot process the request	$\Rightarrow$ <pre>▼ { }</pre>

Try this operation

PATCH /payments/{paymentID}

## Summary

Update a Payment by ID

## Parameters

Name	Located in	Description	Required	Schema
paymentID	path	Payment ID	Yes	⇔ string
payment	body	The payment to be updated	Yes	⇔ <pre>▼model.Payment {   attributes:      ►model.Attributes { }   id:              string   organisation_id: string   type:            string   version:         integer }</pre>

## Responses

Code	Description	Schema
201	Empty result, it is not the created object" TODO fix this	⇔ <pre>▼model.Payment {   attributes:      ►model.Attributes { }   id:              string   organisation_id: string   type:            string   version:         integer }</pre>
404	Can not find ID	⇔ <pre>▼ { }</pre>
500	Cannot process the request	⇔ <pre>▼ { }</pre>

Try this operation

# Models

## model.Attributes

```
▼model.Attributes {  
  amount: string  
  beneficiary_party: ▶model.Party { }  
  charges_information: ▶model.ChargesInformation { }  
  currency: string  
  debtor_party: ▶model.Party { }  
  end_to_end_reference: string  
  fx: ▶model.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: ▶model.Party { }  
}
```

## model.ChargesInformation

```
▼model.ChargesInformation {  
  bearer_code: string  
⇒ receiver_charges_amount: string
```

```

    receiver_charges_currency: string
    sender_charges:           ►[]
}

```

## model.Fx

```

▼model.Fx {
    contract_reference: string
    exchange_rate:      string
⇒    original_amount:   string
    original_currency:  string
}

```

## model.Party

```

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

```

## model.Payment

```

▼model.Payment {
    attributes:          ►model.Attributes { }
    id:                  string
⇒    organisation_id:   string
    type:                string
}

```



```
    version:      integer
  }
```

## model.SenderCharges

```
    ▼model.SenderCharges {  
⇒    amount:  string  
    currency: string  
    }
```