
Order API 3.0

Documentation generated on 2020-05-18.

1. [Overview](#)
2. [Create order](#)
3. [Create payment](#)
4. [Attachment subTypeCode options](#)
5. [Error handling](#)
6. [App support notes](#)

Overview

This API provides methods for self-ordering: create order and create payment.

Version information

Version : v3.0

URI scheme

The POS Cloud is used as hostname

BasePath : /api/order/v3.0

Headers:

X-Token : This token can be obtained in the Cloud

X-Business-Units : The business unit id is the cloud ID (sometimes also referred to as locationId). Since order data is tied to each business unit id, the correct id must be used. This header accepts a single business unit only. The token used should allow access to the given business unit. This can be checked in the API configuration menu of the Cloud.

Create order endpoint

Available from app version 2.42

Method: POST

Path: /orders

Headers: X-Token, X-Business-Units

Request:

```
{
  "tableId": 12,
  "party": {
    "id": 1,
    "name": "d2639fbf-128f-4f3d-a82a-df73b978809a"
  },
  "waiterId": 123,
  "operationUuid": "ed524ec1-4e45-460f-8b31-95ec59e01a70",
  "sales": [
    {
      "itemSku": 123456,
      "isToGoFlag": false,
      "quantity": 11000,
      "regularUnitPrice": 2000,
      "itemName": "Long item name",
      "shortItemName": "Short item name",
      "constraints": [
        {
          "itemSku": 123456,
          "isToGoFlag": false,
          "quantity": 11000,
          "regularUnitPrice": 2000,
          "itemName": "Long item name",
          "shortItemName": "Short item name",
        }
      ]
    }
  ],
  "attachments": []
}
```

Field	Field Type	Value Type	Description
tableId	required	integer	the table the order will be made on. For deliveries, this should be configured on the Cloud
party	required	object	either one of them can be provided, or both (they have to match), or none (pick any free party)
party .id	optional	integer	
party .name	optional	string	if provided, this should be a unique string (see App support notes for 2.42)
waiterId	required	integer	
operationUuid	optional	string	optional uuid4 string, see Avoiding problems with retransmissions
sales	required	array	
sales .itemSku	required	integer	
sales .isToGoFlag	required	boolean	“true” for deliveries, “false” for in-house self-orders
sales .quantity	required	integer	the price should be multiplied by 1000 (2 should be 2000) (see App support notes for 2.42)
sales .regularUnitPrice	optional	integer	the price should be multiplied by 1000 (2€ should be 2000); required only for free price articles, otherwise optional (see App support notes for 2.42)
sales .itemName	optional	string	required only for free text articles, otherwise optional; max allowed length: 60 chars
sales .shortItemName	optional	string	required only for free text articles, otherwise optional; max allowed length: 60 chars
sales .constraints	optional	array	array of dictionaries (recursive structure, one level deep)
attachments	optional	array	see Attachment subTypeCode options

Response:

HTTP Status: 200

```
{
  "tableId": 12,
```

```
"party": {  
  "id": 1,  
  "name": "d2639fbf-128f-4f3d-a82a-df73b978809a"  
}  
}
```

Create payment endpoint

Available from app version 2.42

Method: POST

Path: /payments

Headers: X-Token, X-Business-Units

Request:

```
{
  "tableId": 12,
  "party": {
    "id": 1,
    "name": "d2639fbf-128f-4f3d-a82a-df73b978809a"
  },
  "waiterId": 123,
  "operationUuid": "ed524ec1-4e45-460f-8b31-95ec59e01a70",
  "payment": {
    "tenderId": 123,
    "tipAmount": 22000,
    "appliedToTransactionAmount": 330000
  },
  "attachments": []
}
```

Field	Field Type	Value Type	Description
tableId	required	integer	the table the order will be made on. For deliveries, this should be configured on the Cloud
party	required	object	either a party id or a party name has to be provided (or both)
party .id	optional	integer	required if party name is not provided
party .name	optional	string	required if party id is not provided (see App support notes for 2.42)
waiterId	required	integer	

Field	Field Type	Value Type	Description
operationUuid	optional	string	optional uuid4 string, see Avoiding problems with retransmissions
payment	required	object	
payment .tenderId	required	integer	configured on the cloud, indicates cash or internal payment or partner-specific payment etc.
payment .tipAmount	optional	integer	the amount should be multiplied by 1000 (2€ should be 2000) (see App support notes for 2.42)
payment .appliedTo TransactionAmount	required	integer	everything but tip, no change; the amount should be multiplied by 1000 (2€ should be 2000)
attachments	optional	array	see Attachment subTypeCode options

Response:

HTTP Status: 200

{ }

Attachment SubTypeCode options

Coming soon (see [App support notes](#) for 2.42)

SubTypeCode	Description
01	Additional print tags
02	Arbitrary order metadata

Attachment items expected structure:

For subTypeCode=01:

```
{
  "subTypeCode": "01",
  "blob": {
    "key": "any_tag_name",
    "type": "lines",
    "payload": [
      "Line 1",
      "Line 1",
      "Line 1",
      "Line 1"
    ]
  }
}
```

Field	Field Type	Value Type	Description
subTypeCode	required	string	"01"
blob	required	object	
blob .key	optional	string	name of the tag, written in lower_snake_case
blob .type	optional	string	currently only "lines" is supported
blob .payload	optional	array of strings	each line to print should be a separate string in this array

For subTypeCode=02:

```
{  
  "subTypeCode": "02",  
  "blob": "asdadaASDASDASD=="  
}
```

Field	Field Type	Value Type	Description
subTypeCode	required	string	"02"
blob	required	string	any additional information can be passed in blob as a base64 encoded string

Error handling

200 - OK (order or payment has been processed by the POS, return value returned)

400 - Bad Request for any request that is not valid (missing mandatory fields or unexpected value types)

403 - Unauthorized (check security token or other auth issues)

404 - Not Found (requested business, tenant or URL is absent on the server side).

500 - Server error on any unexpected error (exception).

All errors have a similar body-structure in response, for example:

Errors upon verification:

```
{
  "result":
    {
      "status_code": 400,
      "details": "Missing required field: tableId"
    }
}
```

Avoiding problems with retransmissions

This API establishes a direct connection to the POS. This means internet connection issues at the business can cause timeouts on the API. In such cases, it is advised to retry the operation (order or payment) and indicating that the request is being retried to avoid duplicate orders/payments.

Errors when connecting to the POS contain the `operationUuid` field. This way, the order can be resent with the same `operationUuid`, indicating the connection between the original request and the new one.

```
{
  "result":
    {
      "status_code": 400,
      "details": "Error when connecting to the POS",
      "operationUuid": "ed524ec1-4e45-460f-8b31-95ec59e01a70"
    }
}
```

The `operationUuid` is a UUIDv4 string. It is advisable to leave it null (or not provide it at all) unless the operation is being retried as a result of such an error (by reading the value from the error response). It can also be useful to log that value to troubleshoot potential errors.

App support notes

This section contains implementation notes for different versions of the POS.

App Version 2.42

Implementation notes:

- Fractions of measurement articles not supported: When ordering measurement articles through the Order API, quantity is rounded down to the nearest whole unit. For example, it is not possible to order 2.5L ("`quantity`": 2500) of juice; only 2L or 3L would be accepted.
- Price levels are not supported: The payment API will only take into account the standard price and will fail to settle the correct amount based on price levels.
- Party names are not supported: The party name is accepted and basic validation is made to ensure it is future-proof. However, the party id still needs to be specified in the payment API (for instance). For example, if it is specified, appropriate checks are made to ensure the party being used is free or occupied.
- Attachments are not supported: If they are provided in the API call, they will be ignored.

Known issues:

- User must log in before paying: When the app is closed and reopened, the user must log in before orders can be paid for using the `/payments` endpoint- otherwise a 400 is returned with the misleading message that the party ID is occupied.
- Tip amount has to be specified (can be zero)