Veloce WebConnect interface specifications

System	Veloce POS
Version	1.02 - Adding more responses detail 1.03 - Revenue Centers & Concepts 1.04 - Void/Cancel query order added 1.05 - Added information about Kiosk order 1.06 - Added the selector field in the request examples 1.07 - Added the oper:11 to retrieve selectors configuration
Revision	November 2025
Release	PUBLIC NDA

Table of contents

Revision History

Version	Date	Description of Changes
1.02	Feb 1, 2024	• Added the new information provided in Oper 1 when error result 4 occurs (Missing product) • Modified the way the responses lists are build. They provide now more information and code snippets

1.03	Sep 1, 2024	• Added the support for revenue centers in oper:1 • Added oper:9 to retrieve the list of revenue centers • Added oper:10 to retrieve the list of concepts.
1.04	Nov 22, 2024	• Added Void/Cancel order query
1.05	Sep 30, 2025	• Added information about Kiosk order
1.06	Oct 28, 2025	• Added the selector field in the request (oper:1) examples
1.07	Nov 10, 2025	• Added the new oper:11 to retrive selectors configuration

General

The communication between the application and our API must be done using JSON via a TCP/IP socket (for example : 10.10.10.10:5001). The WebConnect module cannot be accessed via the web, only via the local network.

> ⚠ We highly suggest the partner to build their own windows service to communicate with Veloce on a localhost perspective. Forcing the merchant to have a static IP from the ISP and open a port widely, is highly not recommended for security standpoints.
>
> The service will needs to be installed on the main Veloce server.

Query tag : VeloceRequest:{}

Response tag : VeloceResponse:{}
- "Result":"Value of the result"
- "ResultMsg":"Message of the result"

Veloce Security Token (Optional)

The connection between your online ordering application and the Veloce server can be secured with a single token. The token is a configurable parameter in the BackOffice for our WebConnect module.

The following explains how to accept the Token for the query to be accepted:

NOTE: For this example we are using token123 as the security token.

token123 Followed by the GMT date = result X

The date must use the standard HTTP format (<day-name>, <day> <month> <year> <hour>:<minute>:<second> GMT), as shown here: Tue, 25 Aug 2020 07:28:00 GMT

You must take the result (X) and encode it using SHA1 = result Y

Afterwards, encode result (Y) using Base64 = result T

T = The final token that must be sent to Veloce in the HTTP header under the Control field.

Example of the HTTP header with the token:

GET /VeloceTest HTTP/1.1
Host: 10.10.10.100
Date: Sat, 28 Nov 2009 04:36:25 GMT
Control: plnSRnNml/+nnRBpnwpVpsrvIno=

List of possible errors for Veloce Security Token:

99 - Invalid request

```
{
   "VeloceResponse":{
      "Result":"99",
      "ResultMsg":"Invalid request"
   }
}
```

500 - Transaction Refused

```
{
   "VeloceResponse":{
      "ResultMsg":"Transaction refused"
   }
}
```

998-AccessDenied / Tokenrefused

The token is invalid. Please validate your structure and the token sent to make sure it respects the Veloce structure.

```
{
   "VeloceResponse":{
   "Result":"998",
   "ResultMsg":"Access denied"
   }
}
```

999 - Order module not present

The merchant's license does not have access to the WebConnect module.

```
{
    "VeloceResponse": {
        "Result": "999",
        "ResultMsg": "Order module not present"
    }
}
```

Place an Order (Oper: 1)

Use this query to place an order.

- The query must begin with the `VeloceRequest {}` tag.
- Then you must enter the operation number `(Oper: 1)` .

`Client:{}`

The customer's information. Optional for the TakeOut type but mandatory for the Delivery type with at leastTel, Name, StNumber and Street (the mandatory fields). The information fields available for use with the Delivery type are described below:

- `Tel` - STRING[10] - The customer's phone number. The phone number is usually 10 characters (e.g.: 418999 Mandatory for Delivery type.
- `Name` STRING[24] - The customer's name. 24 characters maximum. Mandatory for Delivery type.
- `StNumber` - The customer's street number. Mandatory for Delivery type.
- `Street` - The customer's street name; limited to 24 characters maximum. Mandatory for Delivery type.
- `App` - The customer's apartment number, if applicable. Optional for Delivery type.
- `t` - The customer's city; limited to 24 characters maximum. Optional for Delivery type.
- `PostalCity` - The customer's postal code. Optional for Delivery type.

`Order:{}`

Array containing the whole order information

- `Type` - The order type. TakeOut or Delivery.
- `ExpectedTime` - The pre-order time (optional). If an expected time is not set, nothing special will print on the invoice and the invoice will not have an ExpectedTime printed on it. The expected time is applicable only for the current day. Currently, we do not manage orders for future days, so your online ordering application will have to hold it until the correct day and then you can send all the orders for THAT day at a particular time within the ExpectedTime for the current day appearing on the invoices.
- `Selector` - (optional) The selector for a multi-revenue center environment (Selector: 1). The selector can be a number from 1 to 20. Each selector can correspond to a different revenue center, and will redirect to a specific workstation/employee. If the selector is omitted, the value 1 is used by default.

`Seats:[]`

Each customer must have their own array of items corresponding to their seat.

`Items:[]`

The array containing all item specifications. Specifically:

- `Product` - Product ID that must match between Veloce and the online ordering application.
- `Quantity` - The item quantity.
- `UnitPrice` - The individual product's price (this is optional, as the Veloce price will be used if the UnitPrice field is not sent for the item).
- `Text` - The product's name (this is optional, as the Veloce product name will be used if the Text field is not ser for the item); limited to 24 characters maximum.

- `RevenueCenter` - Revenue Center ID to use for this item (this is optional, if sending 0 or it is left unspecified, the revenue center associated to the employee/workstation of the selector will be used). (Veloce version 9.83+)

`Invoice:{}`

You must send this tag if you want to print the invoice.

The taxes to send will differ from the merchant's configuration.

- `Tax1` - The value of tax #1 (optional, Veloce will calculate Tax # 1 if no Tax1 is sent).
- `Tax2` - The value of tax #2 (optional, Veloce will calculate Tax # 2 if no Tax2 is sent).
- `Tax3` - The value of tax #3 (optional, Veloce will calculate Tax # 3 if no Tax3 is sent).
- `Tax4` - The value of tax #4 (optional, Veloce will calculate Tax # 4 if no Tax4 is sent).
- `Tax5` - The value of tax #5 (optional, Veloce will calculate Tax # 5 if no Tax5 is sent).

`Payments:[]`

You must send this array if you wish to send the invoice as paid. The maximum number of available payment methods is 4.

If no payment is sent, the order will remain unpaid in the system. This informs the cashier that the order needs to be paid at the counter.
For a delivery order, you must always send a payment mode. If the order is not paid on the web platform but will be paid at the door, you need to send a payment to make sure the order is ready to be delivered, even if it has not yet been paid. We highly suggest to send the payment 1. This will temporary close the invoice in 'Cash' to make it available for Pay at the door terminals.

- `Media` - The number representing the payment method to use in Veloce (configurable according to the customer)
- `Amount` - The transaction total that must be sent to Veloce. This total must equal or exceed the transaction total made on the online ordering application. If the amount is higher than the online ordering application, it will often be used as a tip. In the event that the amount sent is less than the transaction total from the online ordering application, we will make up the difference in money in Veloce.
- For the tip, simply add the tip amount in the total, and Veloce will split the tip.
  - i.e: Total of items with taxes is $50.00, and the client left $5.00 of tip on the web order. The payment amount should be 55.00

TakeOut query

In the following example, the item(s) total is $4.95, the taxes calculated are Tax1 set to 5% and Tax2 set to 9.975%.

- The total of the check taxes included is $5.69
- The first `Media:` 15 with 1 as the amount refers to loyalty program or a gift card redemption with $1 as amount. This is useful if your platform manages loyalty points and/or gift card redemption

- The second `Media`: 19 with 9.69 as amount refers to the rest to pay after the redemption with tip included ($9.69)
- Transaction total: $5.69
- Total redeemed from gift or loyalty : $1.00
- Updated total to pay : $4.69
- Tip added to the transaction : $5.00
- Total amount to send for the second payment: $9.69

Copy

```
{
   "VeloceRequest":{
      "Oper":1,
      "Client":{
         "Tel":"4185555555",
         "Name":"JohnSmith"
      },
      "Order":{
         "ID":1234,
         "Selector":1,
         "Type":"TakeOut",
         "ExpectedTime":"12:00",
         "Seats":[
            {
               "Items":[
                  {
                     "Product":100,
                     "Quantity":1,
                     "UnitPrice":4.95,
                     "Text":"HAMBURGER"
                  },
                  {
                     "Product":200,
                     "Quantity":1,
                     "Text":"KETCHUP"
                  },
                  {
                     "Product":201,
                     "Quantity":1,
                     "Text":"MUSTARD"
                  }
               ],
         "Invoice":{
            "Tax1":0.3,
            "Tax2":0.5,
            "Payments":[
               {
                  "Media":15,
                  "Amount":1
               },
               {
                  "Media":19,
                  "Amount":9.69
               }
            ]
         }
      }
   ]
  }
 }
}
```

Delivery query

In the following example, the item(s) total is $4.95, the taxes calculated are `Tax1` set to 5% and `Tax2` set to 9.975%.

- The total for the check with taxes included is $5.69
- The total amount includes a $5 tip, so it appears as $10.69
- The `Media:` 19 refers to the Web Payment tender on the Veloce backend

```
{
   "VeloceRequest":{
      "Oper":1,
      "Client":{
         "Tel":"4185555555",
         "Name":"JohnSmith",
         "StNumber":"1234",
         "Street":"TestStreet",
         "Appt":"301",
         "City":"Quebec",
         "Postal":"G2G2G2"
      },
      "Order":{
         "ID":1234,
         "Selector":1,
         "Type":"Delivery",
         "ExpectedTime":"12:00",
         "Seats":[
            {
               "Items":[
                  {
                     "Product":100,
                     "Quantity":1,
                     "UnitPrice":4.95,
                     "Text":"HAMBURGER"
                  },
                  {
                     "Product":200,
                     "Quantity":1,
                     "Text":"KETCHUP"
                  },
                  {
                     "Product":201,
                     "Quantity":1,
                     "Text":"MUSTARD"
                  }
               ],
            "Invoice":{
               "Tax1":0.3,
               "Tax2":0.5,
               "Payments":[
                  {
                     "Media":19,
                     "Amount":10.69
                  }
               ]
            }
         }
      ]
   }
}
}
```

Copy

TakeOut query using client record from Oper:2

In the following example, the item(s) total is $4.95, the taxes calculated are `Tax1` set to 5% and `Tax2` set to 9.975%.

- The total for the check with taxes included is $5.69
- The amount total includes a $5 tip, so it appears as $10.69
- The `Media:` 19 refers to the Web Payment tender on the Veloce backend

```
{
   "VeloceRequest":{
      "Oper":1,
      "Client":{
         "Record":"2092"
      },
      "Order":{
         "ID":1234,
         "Selector":1,
         "Type":"TakeOut",
         "ExpectedTime":"12:00",
         "Seats":[
            {
               "Items":[
                  {
                     "Product":100,
                     "Quantity":1,
                     "UnitPrice":4.95,
                     "Text":"HAMBURGER"
                  },
                  {
                     "Product":200,
                     "Quantity":1,
                     "Text":"KETCHUP"
                  },
                  {
                     "Product":201,
                     "Quantity":1,
                     "Text":"MUSTARD"
                  }
               ],
         "Invoice":{
            "Tax1":0.25,
            "Tax2":0.49,
            "Payments":[
               {
                  "Media":19,
                  "Amount":10.69
               }
            ]
         }
      }
   ]
   }
  }
}
```

TakeOut query with Revenue centers

This option is used to specify a revenue center to the POS for every item ordered.

If you do not specify a revenue center, the POS will use the default one based on the backend configuration.

In the following example, the item(s) total is $21.89, the taxes calculated are `Tax1` set to 5% and `Tax2` set to 9.975%.

- The total for the check with taxes included is $25.16
- The amount total includes a $10.00 tip, so it appears as $35.16
- The `Media: 19` refers to the Web Payment tender on the Veloce backend
- `Product items 100, 200 and 201 will be flagged in Revenue Center #1`
- `Product items 300 and 301 will be flagged in Revenue Center #2`

```
{
    "VeloceRequest":{
    "Oper":1,
    "Client":{
            "Record":"2092"
        },
        "Order":{
            "ID":1234,
            "Selector":1,
            "Type":"TakeOut",
            "ExpectedTime":"12:00",
            "Seats":[
            {
                    "Items":[
                        {
                            "Product":100,
                            "Quantity":1,
                            "UnitPrice":4.95,
                            "Text":"HAMBURGER",
                            "RevenueCenter":1
                        },
                        {
                            "Product":200,
                            "Quantity":1,
                            "Text":"KETCHUP"
                            ,
                        },    "RevenueCenter":
                        {     1
                            "Product":201,
                            "Quantity":1,
                            "Text":"MUSTARD"
                            ,
                        },    "RevenueCenter":
                        {     1
                            "Product":300,
                            "Quantity":1,
                            "UnitPrice":14.95,
                            "RevenueCenter":2
                        },
                        {
                            "Product":301,
                            "Quantity":2,
                            "UnitPrice":1.99,
                            "RevenueCenter":
                        }     2
                    ],
                    "Invoice":{
                        "Tax1":1.09,
                        "Tax2":2.18,
                        "Payments":[
                            {
                                "Media":19,
                                "Amount":35.16
            }
          ]
        }
      }
    ]
  }
 }
}
```

Void/Cancel order query

This function is used to void an order from the WebConnect.

It is not possible to void/cancel the original order/check sent, but when a void query is received, it will create the new check in negative to balance the sales.

The order could also be voided manually on the POS.

These fields need to be sent in negative for the POS to engage a void transaction.

In Items[], the Quantity of each item needs to be in negative.

In Invoice[], the Tax1,Tax2,Tax3,Tax4,Tax5 amounts need to be in negative
(only send the used taxes)

In Payments[], the Amount needs to be in negative.

Copy

```
{
   "VeloceRequest":{
      "Oper":1,
      "Client":{
         "Tel":"4185555555",
         "Name":"JohnSmith"
      },
      "Order":{
         "ID":1234,
         "Selector":1,
         "Type":"TakeOut",
         "ExpectedTime":"12:00",
         "Seats":[
            {
               "Items":[
                  {
                     "Product":100,
                     "Quantity":-1,
                     "UnitPrice":4.95,
                     "Text":"HAMBURGER"
                  },
                  {
                     "Product":200,
                     "Quantity":-1,
                     "Text":"KETCHUP
                  }, "
                  {
                     "Product":201,
                     "Quantity":-1,
                     "Text":"MUSTAR
                  } D"
               ],
         "Invoice":{
            "Tax1":-0.3,
            "Tax2":-0.5,
            "Payments":[
               {
                  "Media":15,
                  "Amount":-
               }, 1
               {
                  "Media":19,
                  "Amount":-9.69
               }
            ]
         }
      }
   ]
  }
 }
}
```

Kiosk order

If the order is taken on a Kiosk, there is no difference in between a kiosk order and or takeout order for Veloce.

The field Type to use is also TakeOut

We highly suggest to send the kiosk order identification number in the field ID in the request to match the kiosk order with the POS invoice.
The bracket for Client is not mandatory.

In the following example, the item(s) total is $4.95, the taxes calculated are `Tax1` set to 5% and `Tax2` set to 9.975%.

- The total of the check taxes included is $5.69

- The first `Media: 15` with 1 as the amount refers to loyalty program or a gift card redemption with $1 as amount. This is useful if your platform manages loyalty points and/or gift card redemption
- The second `Media: 19` with 9.69 as amount refers to the rest to pay after the redemption with tip included ($9.69)
- Transaction total: $5.69
- Total redeemed from gift or loyalty : $1.00
- Updated total to pay : $4.69
- Tip added to the transaction : $5.00
- Total amount to send for the second payment: $9.69

Copy

```
{
    "VeloceRequest":{
        "Oper":1,
        "Order":{
            "ID":1234,                  //Use this field for the Kiosk order ID.
            "Selector":1,
            "Type":"TakeOut",
            "Seats":[
                {
                    "Items":[
                        {
                            "Product":100,
                            "Quantity":1,
                            "UnitPrice":4.95,
                            "Text":"HAMBURGER"
                        },
                        {
                            "Product":200,
                            "Quantity":1,
                            "Text":"KETCHUP"
                        },
                        {
                            "Product":201,
                            "Quantity":1,
                            "Text":"MUSTARD"
                        }
                    ],
            "Invoice":{
                "Tax1":0.3,
                "Tax2":0.5,
                "Payments":[
                    {
                        "Media":15,
                        "Amount":1
                    },
                    {
                        "Media":19,
                        "Amount":9.69
                    }
                ]
            }
        }
    ]
  }
}
```

List of responses:

0 - Order posted

The order was successfully processed in the POS.

```
{
    "VeloceResponse": {
        "Oper": "1",
        "Result": "0",
        "ResultMsg": "Order posted",
        "ID": YOUR_PROVIDED_ID,
        "OrderNumber": VELOCE_ORDER_ID
```

```
    }
}
```

1 - Order type missing or invalid

The `Type` field is missing or contains an invalid type.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"1",
        "ResultMsg":"Order type misisng or invalid",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
}
```

2 - Internal Error

This error should never happen. If you receive this error, please contact our POS support team.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"2",
        "ResultMsg":"Internal error"
    }
}
```

3 - Multiple client match

This error occurs when there are too many clients matching the information provided in the order.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"3",
        "ResultMsg":"Multiple client match"
    }
}
```

To avoid this problem, we highly suggest to use the oper: 2 (Query the clients database) to retrieve the client's unique record ID to send it in the order. This way, you will avoid receiving a Result 3.

4 - At least one product not found

One of the products in the order does not exist in the Veloce database.

In Veloce version 9.77.7 (and later), there is an option to include the missing product ID in the response.
If the merchant is running a version below the requirement, the Product field will not be included as part of the response.
If the merchant is running the minimum version, make sure that the option to add missing product in the response is enabled in Veloce backend.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"4",
        "ResultMsg":"At least one product not found",
        "ID":YOUR_PROVIDED_ID,
        "Product":3313
    }
}
```

5 - Missing item fields

The `Items` bracket is not included in the request, or the structure is invalid.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"5",
        "ResultMsg":"Missing item fields",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
}
```

Validate your JSON structure to make sure the `Items` bracket is valid.

6 - No item specified

No item(s) specified in the `Items` bracket

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"6",
        "ResultMsg":"No item specified",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
}
```

Validate there is at least one item in the `Items` bracket.

7 - Missing customer field

In the event where the request was sent by a non-existing client, Veloce will create a new customer at the same time, but the information received is insufficient.

The following fields are mandatory to allow Veloce to create the client:

```
"Name":"",
"Tel":"",
"StNumber":"",
"Street":"
```

8 - Order not posted (unknown error)

This error can occur only with the TakeOut type. It is caused by the table section in the Backoffice that does not have enough tables to manage multiple orders at the same time.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"8",
        "ResultMsg":"Order not posted (unknown error)",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
}
```

To avoid this error, ask Veloce support to configure the Section#30 in the Backoffice with more than 1 table available.

9 - Error with client record

This error will occur when the client record sent in the request does not exist in the database.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"9",
        "ResultMsg":"Error with client record"
    }
}
```

10 - Order posted but need to be confirmed because of a problem with the POS

The Veloce workstation is not open but the order has been processed by the Veloce server

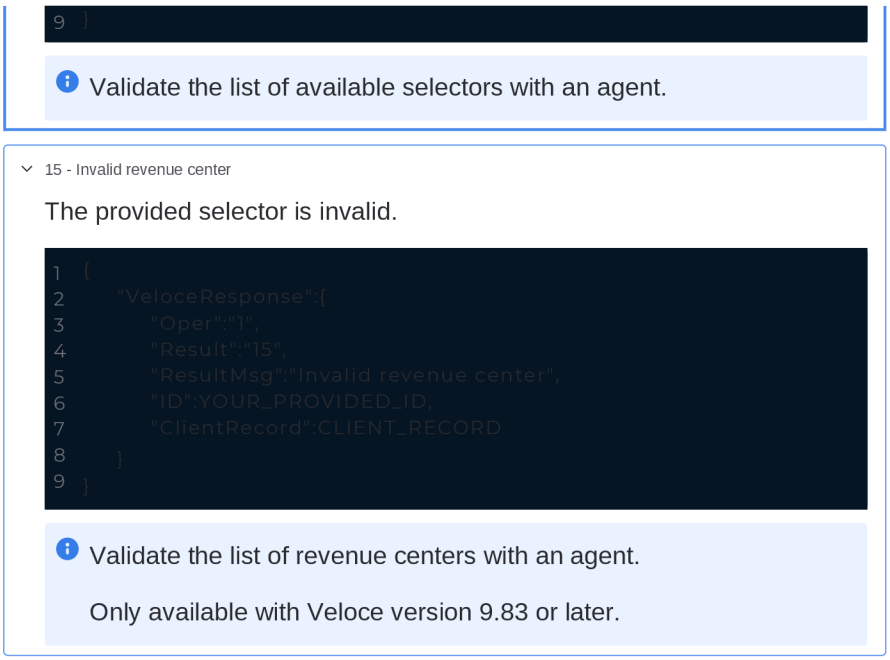
```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"10",
        "ResultMsg":"Order posted but need to be confirmed because of a problem with POS",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
}
```

Make sure the POS configured to process web orders is operating and online.

14 - Invalid selector.

The provided selector is invalid.

```
{
    "VeloceResponse":{
        "Oper":"1",
        "Result":"14",
        "ResultMsg":"Invalid selector",
        "ID":YOUR_PROVIDED_ID,
        "ClientRecord":CLIENT_RECORD
    }
```



Query the clients database (Oper: 2)
This query allows you to query the clients database to verify if the client already exists while avoiding the “Multiple client match” error when there are too many clients with the same information in the database.

- If the client exists, Veloce will respond with the record number.
- You can use the record number instead of sending all of the client’s information when posting a new order.
- We suggest querying the clients database before sending an order to Veloce (Oper: 1) and using the record number when posting the order.

The mandatory fields to query the clients database are “Name” and “Tel”.

Query examples with multiple fields

```
{
   "VeloceRequest":{
      "Oper":2,
      "Name":"John Smith",
      "Tel":"4185555555",
      "Street":"Test Street",
      "StNumber":"1234",
      "Appt":"301",
      "City":"Quebec",
      "Postal":"G2G2G2"
   }
}
```

Copy

Query example with the mininum fields

```
{
   "VeloceRequest":{
      "Oper":2,
      "Name":"John Smith",
      "Tel":"4185555555"
   }
}
```

Copy

List of responses:

1 - Client found

If the client exists using the information provided to Veloce, it will respond with a Result of 1 and the client's information.


```
{
   "VeloceResponse":{
      "Oper":"2",
      "Result":"1",
      "Clients":[
         {
            "Record":24574,
            "Name":"Client Test",
            "Tel":"4185555555",
            "StCode":"",
            "StNumber":1234,
            "Street":"Test Street",
            "Appt":"5",
            "Near":"",
            "City":"Quebec",
            "Postal":"G2G2G2",
            "Fields":[
               "",
               "",
               ""
            ],
            "Specials":[
               "",
               ""
            ],
            "ForcedZone":0
         }
      ]
   }
}
```


0 - Client not found


```
{
   "VeloceResponse":{
      "Oper":"2",
      "Result":"0",
      "ResultMsg":"Client not found"
   }
}
```

POS Status (Ping) (Oper: 3)

Use this query to check the status of the POS that manages the online orders.

- The query must begin with the VeloceRequest {} tag.
- Then you must enter the operation number (Oper: 3).

Query example:


```
{
   "VeloceRequest":{
      "Oper":3
   }
}
```


Copy

List of responses:

0-OnLine

The POS is ready to receive web order.

```
{
    "VeloceResponse": {
        "Oper": "3",
        "Result": "0",
        "ResultMsg": "OnLine"
    }
}
```

1 -Internal error

The POS file is locked. We suggest that you reboot the Veloce workstation managing online orders.

```
{
    "VeloceResponse": {
        "Oper": "3",
        "Result": "1",
        "ResultMsg": "Internal error"
    }
}
```

2 - OffLine

The POS is not ready to accept a web order. Make sure the workstation application is running on the machine.

```
{
    "VeloceResponse": {
        "Oper": "3",
        "Result": "2",
        "ResultMsg": "OffLine"
    }
}
```

Status of an Order (Oper: 4)

Use this query to retrieve the status of an order sent to the system.

- The query must begin with `the VeloceRequest {}` tag.
- Then you must enter the operation number `(Oper: 4)`.
- `OrderNumber -` The order number as returned by Veloce WebConnect.
- `OrderType -` The order type. `TakeOut` or `Delivery`. You must specify the `OrderType` because each order type can potentially use the same order number.

Query example for Delivery order:

```
{
    "VeloceRequest":{
        "Oper":4,
        "OrderNumber":1,
        "OrderType":"Delivery"
    }
}
```

Copy

Query example for TakeOut order:

```
{
   "VeloceRequest":{
      "Oper":4,
      "OrderNumber":800,
      "OrderType":"TakeOut"
   }
}
```

Copy

List of responses:

0 - Order not found

The provided order can not be found in the system or it has already been closed.

If a `TakeOut` order is sent with the payment, the order is automatically closed by the POS and it will always send back 'Order not found'

It is only useful when the merchant is using `Delivery` with the dispatch system and/or kitchen display system.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "0",
        "ResultMsg": "Order not found"
    }
}
```

1 - Invoice not printed

The order has been received and is in the WebConnect queue, but the order has not been printed yet.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "1",
        "ResultMsg": "Invoice not printed"
    }
}
```

2 - Invoice printed

The order has been received by the WebConnect and processed by the POS.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "2",
        "ResultMsg": "Invoice printed"
    }
}
```

3 - Ready to go

The order has been prepared in the kitchen.

This option is only available with Kitchen Display System

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "3",
        "ResultMsg": "Order ready to go"
    }
}
```

4 - Order gone

The order has left the restaurant with a driver.

> This option is only available with `Delivery` module in Dispatch mode.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "4",
        "ResultMsg": "Order gone"
    }
}
```

10 - Internal error

This error should never happen. If you receive this error, please contact our POS support team.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "10",
        "ResultMsg": "Internal error"
    }
}
```

11 - OrderNumber not present

The order number is not specified.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "11",
        "ResultMsg": "OrderNumber not present"
    }
}
```

12 - OrderType not present

The order type is not specified.

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "11",
        "ResultMsg": "OrderType not present"
    }
}
```

999 - Order module not present

The merchant's license does not have access to the WebConnect module.

```
{
    "VeloceResponse": {
        "Result": "999",
        "ResultMsg": "Order module not present"
    }
}
```

Get the List of Items (Oper: 5)

Use this query to retrieve the complete list of programmed items.

The query must begin with the `VeloceRequest {}` tag.

Then you must enter the operation number (Oper: 5).

- `StartFrom` - Optional - The position from which to begin retrieving items from the item list. For example, 1 is the first item in the list whereas 14 would be the 14th list item.
- `RevenueCenters` - Optional - The revenue centers information. 0 (don't show) or 1 (show). This will display the revenue centers information of the item (Price override and Availabilities by revenue center).
- The query can only return 200 items maximum at a time; If the response size can't fit 200 items, it will stop before the 200 limit. If the list of items returned is greater than 200, a `Continue` tag appears with a numeric value to indicate the next `StartFrom` position (for example, `"Continue": "156"` means the next Oper: 5 request should use 156 as the `StartFrom` value). If the `Continue` tag returns No, there are no additional items in the list to retrieve.

Query example to retrieve all items starting from the first.

Copy

```
{
   "VeloceRequest":{
      "Oper":5,
      "StartFrom":1,
      "RevenueCenters":0
   }
}
```

List of responses:

1 - List of items

If you receive the `Continue` tag with a numeric value, you must resend the Oper: 5 request with the `StartFrom` at the number indicated in the `Continue` tag until you receive a `Continue` tag with the value `No`, indicating there are no more items to retrieve.

- With `"RevenueCenters":0` or without `"RevenueCenters"`

```
{
    "VeloceResponse": {
        "Oper": "5",
```

```
        "Result": "1",
        "Items": [{
                "Number":654,
                "Description": "HOT DOG",
                "PreparationTime": 0,
                "RegularPrice": "1.99",
                "Division": {
                "Number":    1,
                "Description":       "ENTREES 2"
                }, "BigDivision": {
                0,
                    "Number":
                    "Description": "Nourriture"
                },
                "Price": {
                    "Mode": [{
                            "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                        }
                    ]
                },
                "Tax1":  1,
                "Tax2":  1,
                "Tax3":  0,
                "Tax4":  0
            }
        ],
        "Continue": "No"
    }
}
```


- With `"RevenueCenters":1`


```
{
    "VeloceResponse": {
        "Oper": "5",
        "Result": "1",
        "Items": [{
                "Number":654,
                "Description":  "HOT DOG",
                "PreparationTime": 0,
                "RegularPrice": "90000.99",
                "Division": {
                "Number":    1,
                "Description":       "ENTREES 2"
                }, "BigDivision": {
                0,
                    "Number":
                    "Description": "Nourriture"
                },
                "PriceOverrides": [{
                        "RevenueCenter":1,
                        "Price": {
                            "Mode": [{
                                    "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                                }, {
                                    "Level": ["$1.99", "$1.99", "$1.99", "$1.99"]
                                }, {
```

```
                                    "Level": ["$1.99", "$1.99",
"$1.99", "$1.99"]
                                }, {
                                    "Level": ["$1.99", "$1.99",
"$1.99", "$1.99"]
                                }
                            ]
                        }
                    }, {
                        "RevenueCenter": 2,
                        "Price": {
                            "Mode": [{
                                    "Level": ["$1.99", "$1.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$1.99", "$1.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$1.99", "$1.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$1.99", "$1.99",
"$0.00", "$0.00"]
                                }
                            ]
                        }
                    }, {
                        "RevenueCenter": 3,
                        "Price": {
                            "Mode": [{
                                    "Level": ["$2.99", "$2.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$2.99", "$2.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$2.99", "$2.99",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$2.99", "$2.99",
"$0.00", "$0.00"]
                                }
                            ]
                        }
                    }, {
                        "RevenueCenter": 4,
                        "Price": {
                            "Mode": [{
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                }
                            ]
                        }
                    }, {
                        "RevenueCenter": 5,
                        "Price": {
                            "Mode": [{
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                }, {
                                    "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
```

```
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }
                                ]
                            }
                        }, {
                            "RevenueCenter": 6,
                            "Price": {
                                "Mode": [{
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }
                                ]
                            }
                        }, {
                            "RevenueCenter": 7,
                            "Price": {
                                "Mode": [{
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }
                                ]
                            }
                        }, {
                            "RevenueCenter": 11,
                            "Price": {
                                "Mode": [{
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }, {
                                        "Level": ["$0.99", "$0.00",
"$0.00", "$0.00"]
                                    }
                                ]
                            }
                        }, {
                            "RevenueCenter": 14,
                            "Price": {
                                "Mode": [{
                                        "Level": ["$0.00", "$0.00",
"$0.00", "$0.00"]
                                    }, {
```

```
                                    "Level":  ["$0.00",  "$0.00",
    "$0.00", "$0.00"]
                                }, {
                                    "Level":  ["$0.00",  "$0.00",
    "$0.00", "$0.00"]
                                }, {
                                    "Level":  ["$0.00",  "$0.00",
    "$0.00", "$0.00"]
                                }
                            ]
                        }
                    }, {
                        "RevenueCenter": 100,
                        "Price":  {
                            "Mode": [{
                                    "Level":  ["$1.99",  "$1.99",
    "$1.99", "$1.99"]
                                }, {
                                    "Level":  ["$1.99",  "$1.99",
    "$1.99", "$1.99"]
                                }, {
                                    "Level":  ["$1.99",  "$1.99",
    "$1.99", "$1.99"]
                                }, {
                                    "Level":  ["$1.99",  "$1.99",
    "$1.99", "$1.99"]
                                }
                            ]
                        }
                    }
                ],
                "Availabilities":  [{
                        "RevenueCenter": 1,
                        "ModesPerWeekDay":
                            "Mode": [{
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1]
                                }
                            ]
                        },
                        "PerPriceLevel":  {
                            "Level":  [1,  1,  1,  1]
                        }
                    }, {
                        "RevenueCenter": 2,
                        "ModesPerWeekDay":
                            "Mode": [{
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1]
                                }
                            ]
                        },
                        "PerPriceLevel":  {
                            "Level":  [1,  1,  1,  1]
                        }
                    }, {
                        "RevenueCenter": 3,
                        "ModesPerWeekDay":
                            "Mode": [{
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1],
                                    "Day":  [1,  1,  1,  1,  1,  1,  1]
                                }
                            ]
                        },
                        "PerPriceLevel":  {
```

```
                    "Level": [1,    1,  1,  1]
                }
            }, {
                "RevenueCenter": 4,
                "ModesPerWeekDay": {
                    "Mode": [{
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day": [1,   1,  1,  1,  1,  1,  1]
                        }
                    ]
                },
                "PerPriceLevel": {
                    "Level": [1,    1,  1,  1]
                }
            }, {
                "RevenueCenter": 5,
                "ModesPerWeekDay": {
                    "Mode": [{
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day": [1,   1,  1,  1,  1,  1,  1]
                        }
                    ]
                },
                "PerPriceLevel": {
                    "Level": [1,    1,  1,  1]
                }
            }, {
                "RevenueCenter": 6,
                "ModesPerWeekDay": {
                    "Mode": [{
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day": [1,   1,  1,  1,  1,  1,  1]
                        }
                    ]
                },
                "PerPriceLevel": {
                    "Level": [1,    1,  1,  1]
                }
            }, {
                "RevenueCenter": 7,
                "ModesPerWeekDay": {
                    "Mode": [{
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day": [1,   1,  1,  1,  1,  1,  1]
                        }
                    ]
                },
                "PerPriceLevel": {
                    "Level": [1,    1,  1,  1]
                }
            }, {
                "RevenueCenter":  11,
                "ModesPerWeekDay": {
                    "Mode": [{
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day":   [1, 1,  1,  1,  1,  1,  1],
                            "Day": [1,   1,  1,  1,  1,  1,  1]
                        }
                    ]
                },
                "PerPriceLevel": {
                    "Level": [1,    1,  1,  1]
```

```
                        }
                    }, {
                        "RevenueCenter": 14,
                        "ModesPerWeekDay": {
                            "Mode": [{
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1]
                                }
                            ]
                        },
                        "PerPriceLevel": {
                            "Level": [1,  1, 1, 1]
                        }
                    }, {
                        "RevenueCenter"100,
                        "ModesPerWeekDay": {
                            "Mode": [{
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1],
                                    "Day": [1, 1, 1, 1, 1, 1, 1]
                                }
                            ]
                        },
                        "PerPriceLevel": {
                            "Level": [1,  1, 1, 1]
                        }
                    }
                ],
                "Price": {
                    "Mode": [{
                            "Level": ["$1.99", "$1.99", "$1.99",
    "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99",
    "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99",
    "$1.99"]
                        }, {
                            "Level": ["$1.99", "$1.99", "$1.99",
    "$1.99"]
                        }
                    ]
                },
                "Tax1": 1,
                "Tax2": 1,
                "Tax3": 0,
                "Tax4": 0
            }
        ],
        "Continue": "No"
    }
}
```


999 - Order module not present

The merchant’s license does not have access to the WebConnect module.

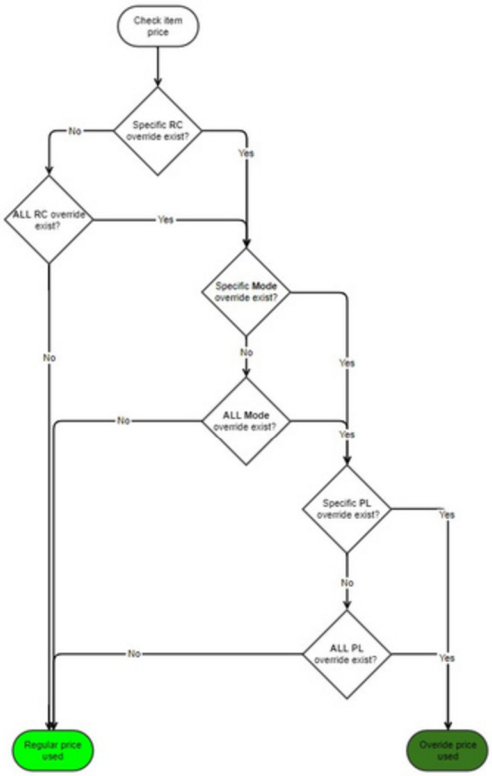

```
{
    "VeloceResponse": {
        "Oper": "4",
        "Result": "999",
        "ResultMsg": "Order module not present"
```

```
6       }
7   }
```

⚠ On version 9.83 and later, the list of price exceptions replaces the now-deprecated (mode, price level) pricing grid.

For the “RevenueCenter”, “Mode” and “PriceLevel” fields, a value of 0 is considered as “unspecified”/”all”.

The behavior is as follows:



Create a new street (Oper: 6)

Use this query to create a new street in the Veloce Streets database when the merchant is using ‘Street Validation’

- The query must begin with `the VeloceRequest {}` tag.
- Then you must enter the operation number `(Oper: 6)`.
- You can configure up to 8 different blocks of street numbers within one street.
  - `Code:` The unique street code
  - `Description:` Input the street name
  - `Zone:` Zone’s number specified by the client itself based on their zone configuration
  - `StNumFrom:` First door number
  - `StNumTo:` Last door number
  - `Fee:` Amount of delivery fee for this block

In the following example, we are creating a new street using CodeTEST and Test Street as the `Description`.

- The street has only one block, from door #10 to #299, with a fee of $5.00

```
{
"VeloceRequest":{
"Oper":6,
"Street":[
      {
           "Code":"TEST",
           "Description":"Test Street",
           "Zone":1,
           "StNumFrom":10,
           "StNumTo":299,
           "Fee":5.00
      }
    ]
  }
}
```

Copy

In the following example, we are creating a new street using the `Code`TEST2 and Test Street2 as the `Description`.

- The street has two blocks separated by 2 zones

```
{
"VeloceRequest":{
"Oper":6,
"Street":[
      {
           "Code":"TEST2",
           "Description":"Test Street2",
           "Zone":1,
           "StNumFrom":10,
           "StNumTo":299,
           "Fee":5.00
      },
      {
           "Code":"TEST2",
           "Description":"Test Street2",
           "Zone":2,
           "StNumFrom":300,
           "StNumTo":599,
           "Fee":7.00
      }
    ]
  }
}
```

Copy

The first block from door #10 to #299 is part of Zone#1 with a fee of $5.00

```
{
  "Code":"TEST2",
  "Description":"Test Street2",
  "Zone":1,
  "StNumFrom":10,
  "StNumTo":299,
  "Fee":5.00
},
```

Copy

The second block from door #300 to #599 is part of Zone#2 with a fee of $7.00

```
{
  "Code":"TEST2",
  "Description":"Test Street2",
  "Zone":2,
  "StNumFrom":300,
  "StNumTo":599,
  "Fee":7.00
}
```

Copy

List of responses:

0-Success

The street was successfully created on the Veloce backend.

```
{
   "VeloceResponse":{
      "Oper":"6",
      "Street":[
         {
            "Code":"TEST",
            "Result":"0",
            "ResultMsg":"Success"
         }
      ]
   }
}
```

If the request included multiple blocks, the response will have a response code for each block.

```
{
   "VeloceResponse":{
      "Oper":"6",
      "Street":[
         {
            "Code":"TEST",
            "Result":"0",
            "ResultMsg":"Success"
         },
         {
            "Code":"TEST",
            "Result":"0",
            "ResultMsg":"Success"
         }
      ]
   }
}
```

4 - Invalid data

The data provided is not valid for use with Veloce. Please validate your data structure.

```
{
   "VeloceResponse":{
      "Oper":"6",      "Result":4,
      "ResultMsg":"Invalid data"

   }
}
```

10-InternalError

Internal Error. Contact Product Support

```
{
   "VeloceResponse":{
      "Oper":"6",
      "Result":10,
      "ResultMsg":"Internal Error"
   }
}
```

Query the streets database (Oper: 7)

Use this query to query and varify if a street already exists using the same code.

- The query must start with `the VeloceRequest {}` tag.
- Then you must enter the operation `number (Oper: 7)`.

```
{
"VeloceRequest":{
    "Oper":7
    "Code":"TEST"
  }
}
```

Copy

List of responses:

0-Success

The street exists in the database, and the response includes all the data.

If there is more than one block, `the Block` array will contain all blocks.

```
{
  "VeloceResponse":{
     "Oper":"7",
     "Result":"0",
     "ResultMsg":"Success",
     "Street":{
        "Code":"TEST",
        "Description":"Test Street",
        "Block":[
           {
              "StNumFrom":10,
              "StNumTo":299,
              "Zone":9,
              "Fee":"1.99"
           }
        ]
     }
  }
}
```

1 - Invalid code

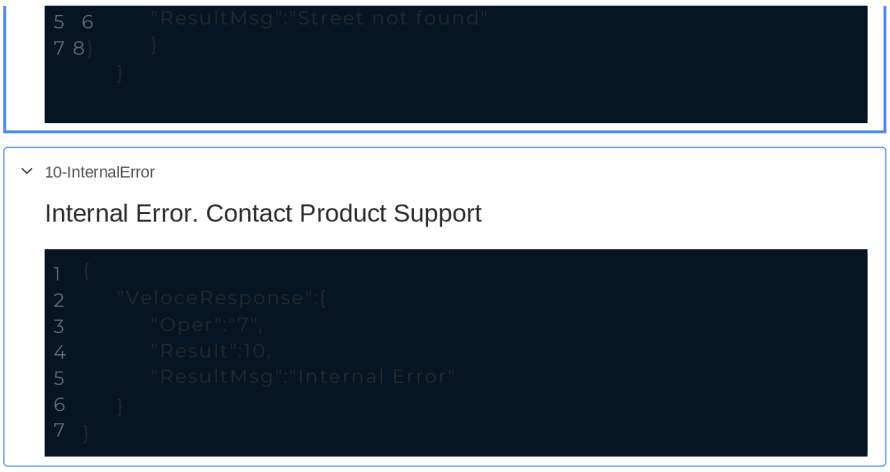
The provided code is invalid.

```
{
  "VeloceResponse":{
     "Oper":"7",
     "Result":"1",
     "ResultMsg":"Invalid code"
     }
  }
}
```

2 - Street not found

No street was found with the code provided.

```
{
  "VeloceResponse":{
     "Oper":"7",
     "Result":"2",
```



Send text to a printer (Oper: 8)

Use this query to send text on a printer installed at the customer.

The query must begin with the VeloceRequest {} tag

Then you must enter the operation number (Oper: 8)

- Printer - The name of the printer configured in Veloce (may vary depending on the customer)
  - Line:[] - The different lines you want to send to the printer in the Print section
    - Text - The text to send to the printer
    - Center - If you want to center the text; a value of 0 will not be centered, a value of 1 will be centered
    - Size - Specific text values: COMPRESSED, LARGE or NORMAL
    - Red - Print in red mode if supported by the printer; a value of 0will not print in red mode, a value of 1 will print in red mode

Query example:

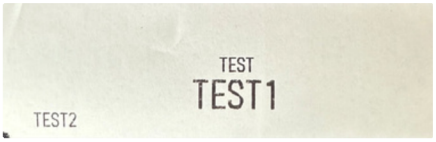
Copy

```
{
   "VeloceRequest":{
      "Oper":8,
      "Printer":"Kitchen",
      "Print":{
         "Line":[
            {
               "Text":"TEST",
               "Center":1,
               "Size":"COMPRESSED",
               "Red":1
            },
            {
               "Text":"TEST1",
               "Center":1,
               "Size":"LARGE",
               "Red":1
            },
            {
               "Text":"TEST2",
               "Center":0,
               "Size":"NORMAL",
               "Red":0
            }
         ]
      }
   }
}
```

Printout example:

This is the result of the query example.

- TEST is COMPRESSED and CENTERED
- TEST1 is LARGE and CENTERED
- TEST2 is NORMAL and NOT CENTERED
- The red option will put the text RED in the kitchen if the merchant uses a ribbon with 2 colors



List of responses:

0 - Successfully added to WS Spooler

In case the information to print is sent to a receipt printer, the response will specify that the request has been added to the WS spooler.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"0",
        "ResultMsg":"Successfully added to WS spooler"
    }
}
```

0 - Successfully added to spooler

In case the information to print is sent to a kitchen printer, the response will specify that the request has been added to the spooler.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"1",
        "ResultMsg":"Successfully added to spooler"
    }
}
```

1 -Noprinterspecified

The printer name is not specified in the request.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"1",
        "ResultMsg":"No printer specified"
    }
}
```

2 - Printer not found

The printer name is not found in the Veloce backend. Please validate the printer's name.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"2",
```

```
    "ResultMsg":"Printer not found"
}
```

3 - Nothing to print

The request contains no line information to print.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"3",
        "ResultMsg":"Nothing to print"
    }
}
```

4 - Error during format

The text to print is not valid. Please validate your request with Product suppor t.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"4",
        "ResultMsg":"Error during format"
    }
}
```

5 - Error adding to spooler

The request is not valid and could not be processed by Veloce print spooler. Contact Product support.

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"5",
        "ResultMsg":"Error adding to WS spooler"
    }
}
```

```
{
    "VeloceResponse":{
        "Oper":"8",
        "Result":"5",
        "ResultMsg":"Error adding to spooler"
    }
}
```

Request available Revenue centers (Oper: 9)

Use this query to query the revenue centers database and return a list of available revenue centers.

- The query must begin with `the VeloceRequest {}` tag
- You must enter the operation number (Oper: 9)

Note:

> This query requires Veloce version 9.86 or later

Query example:

```
{
   "VeloceRequest":{
      "Oper":9
   }
}
```

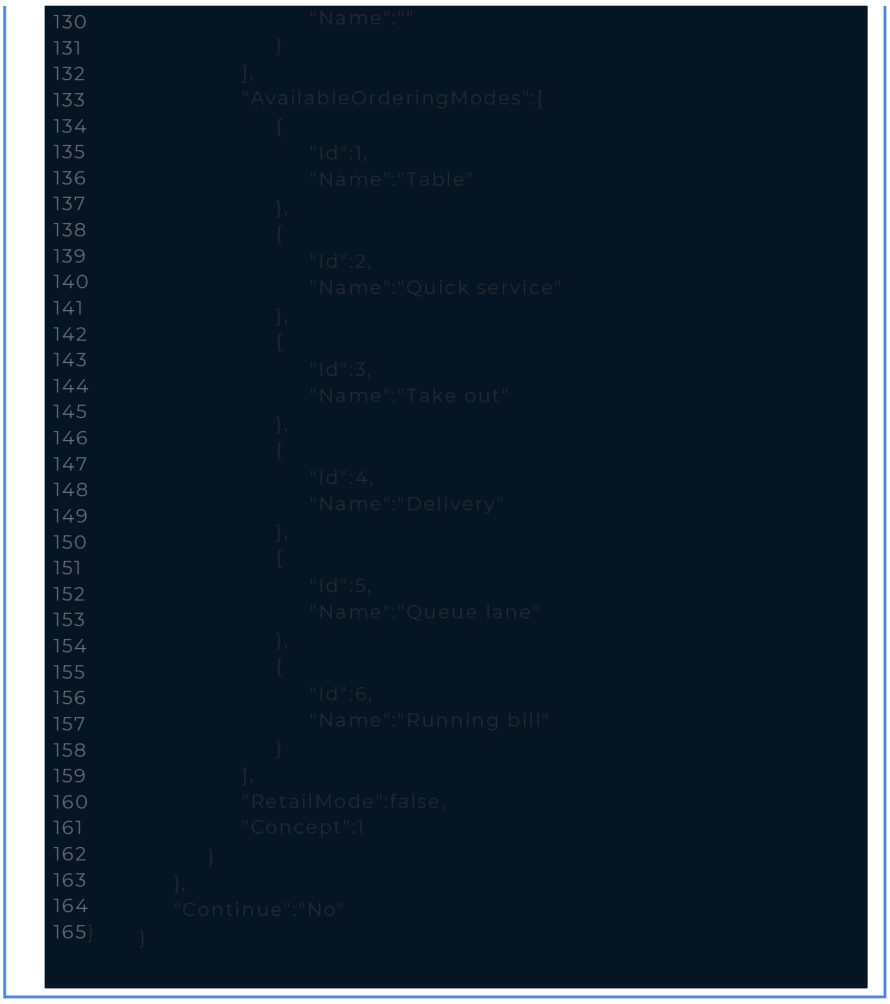
Copy

List of responses:

0 - Receive the list of Revenue centers.


```
{
   "VeloceResponse":{
      "Oper":9,
      "Result":0,
      "RevenueCenters":[
         {
            "Number":1,
            "Description":"UBER EATS",
            "SecondName":"",
            "ExternalId":"",
            "AvailablePriceLevels":[
               {
                  "Id":1,
                  "Name":"SUR PLACE"
               },
               {
                  "Id":2,
                  "Name":"LIVRAISON"
               },
               {
                  "Id":3,
                  "Name":""
               },
               {
                  "Id":4,
                  "Name":""
               }
            ],
            "AvailableOrderingModes":[
               {
                  "Id":1,
                  "Name":"Table"
               },
               {
                  "Id":2,
                  "Name":"Quick service"
               },
               {
                  "Id":3,
                  "Name":"Take out"
               },
               {
                  "Id":4,
                  "Name":"Delivery"
               },
               {
                  "Id":5,
                  "Name":"Queue lane"
               },
               {
                  "Id":6,
                  "Name":"Running bill"
               }
            ],
            "RetailMode":false,
            "Concept":1
         },
         {
```

```
                "Number":2,
                "Description":"DOORDAS
                H",         "SecondName":"",
                "ExternalId":"",
                "AvailablePriceLevels":[
                    {
                        "Id":1,
                        "Name":"SUR PLACE"
                    },
                    {
                        "Id":2,
                        "Name":"LIVRAISON"
                    },
                    {
                        "Id":3,
                        "Name":""
                    },
                    {
                        "Id":4,
                        "Name":""
                    }
                ],
                "AvailableOrderingModes":[
                    {
                        "Id":1,
                        "Name":"Table"
                    },
                    {
                        "Id":2,
                        "Name":"Quick service"
                    },
                    {
                        "Id":3,
                        "Name":"Take out"
                    },
                    {
                        "Id":4,
                        "Name":"Delivery"
                    },
                    {
                        "Id":5,
                        "Name":"Queue lane"
                    },
                    {
                        "Id":6,
                        "Name":"Running bill"
                    }
                ],
                "RetailMode":false,
                "Concept":1
            },
            {
                "Number":3,
                "Description":"SKIP",
                "SecondName":"",
                "ExternalId":"",
                "AvailablePriceLevels":[
                    {
                        "Id":1,
                        "Name":"SUR PLACE"
                    },
                    {
                        "Id":2,
                        "Name":"LIVRAISON"
                    },
                    {
                        "Id":3,
                        "Name":""
                    },
                    {
                        "Id":4,
```



99 - Invalid request

Failed to retrieve the list of revenue centers / Installed POS version not compatible

Request available concepts (Oper: 10)

Use this query to query the concepts database and return a list of available concepts.

- The query must begin with `the VeloceRequest {}` tag
- You must enter the operation number (Oper: 10)

Note:

This query requires Veloce version 9.86 or later

Query example:

```
{
   "VeloceRequest":{
      "Oper":10
   }
}
```

Copy

List of responses:

0 - Receive the list of concepts

```
{
   "VeloceResponse":{
      "Oper":10,
      "Result":0,
      "Concepts":[
         {
            "Number":1,
            "Description":"CAGE"
         },
         {
            "Number":2,
            "Description":"Concept #2"
         },
         {
            "Number":3,
            "Description":"Concept #3"
         },
         {
            "Number":4,
            "Description":"Concept #4"
         }
      ],
      "Continue":"No"
   }
}
```

99 - Invalid request

Failed to retrieve the list of concepts / Installed POS version not compatible

```
{
   "VeloceResponse":{
   "Oper":"10",
   "Result":"99",
   "ResultMsg":"Invalid request"
   }
}
```

Request available WebConnect selectors (Oper: 11)

Use this query to get every configured selectors in the WebConnect configuration.

- The query must begin with `the VeloceRequest {}` tag
- You must enter the operation number (Oper: 11)

Note:

This query requires Veloce version 9.87.10 or later

Query example:

```
{
   "VeloceRequest":{
      "Oper":11
   }
}
```

Copy

List of responses:

0 - Receive the list of configured selectors


```
{
    "VeloceResponse": {
        "Oper":     "11",
        "Result":   "0",
        "Selectors": [
            {
                "Number": 1,
                "Workstation": {
                "Number": 1,
                "Description": "Caisse#1 - Windows"
                },
                "Employee": {
                    "Number": 11,
                    "Description": "Michel Untel"
                },
                "RevenueCenter": {
                    "Number": 1,
                    "Description": "RESTO_Windows"
                }
            },
            {
                "Number": 17,
                "Workstation": {
                    "Number": 17,
                    "Description": "Caisse#17"
                },
                "Employee": {
                    "Number": 17,
                    "Description": "Mandataire"
                },
                "RevenueCenter": {
                    "Number": 6,
                    "Description": "PAX_2"
                }
            }
        ],
        "Continue": "No"
    }
}
```


99 - Invalid request

Failed to retrieve the list of configured selectors / Installed POS version not compatible


```
{
   "VeloceResponse":{
   "Oper":"11",
   "Result":"99",
   "ResultMsg":"Invalid request"
   }
}
```