

Order & Product

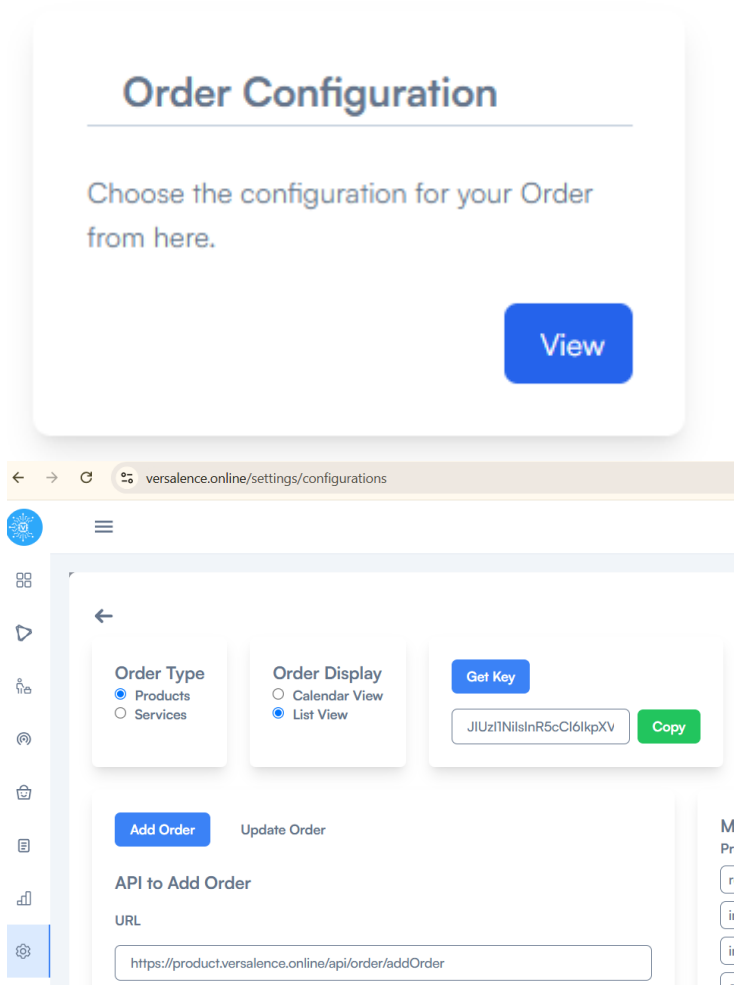
The **Order / Service / Appointment Module** in vCX provides a centralized view of all your external orders, services, and appointments. This feature ensures businesses can seamlessly track, manage, and act on these records without switching systems, enabling faster and smarter workflows.

- [Managing Data in vCX](#)
 - [Insert or update order via API](#)
 - [Update Order Status via API](#)
- [Getting updates from vCX](#)
 - [Adding webhook for order updates](#)

Managing Data in vCX

- **Order creation** Users can create orders using an API, and then send them to a point of sale application for fulfillment.
- **Order management** Users can use an API to modify and cancel pending orders, retrieve daily orders, and more.
- **Supply chain transparency** Users can integrate an API with their Enterprise Resource Planning (ERP) or Truck Management Software (TMS) to get orders into a platform. This can help with supply chain transparency and control.

Insert or update order via API



Add Order into the system

```
https://productbackend.versal.one/api/order/addOrder
{
  // Mandatory Fields
  "company_id": VERPL000972024,
  "company_order_id": "ORD-001",
  "totalAmount": 1500.75,
  "firstName": "John",
  "phoneNumber": "+1234567890",
  "postalCode": "56789",
  "streetAddress1": "123 Main Street",
}
```

```

// Optional Fields
"lastName": "Doe",
"email": "john.doe@example.com",
"streetAddress2": "Apt 4B",
"city": "Springfield",
"state": "Illinois",
"specialInstructions": "Leave package at the front door.",
"products": [
  { "product_id": "101", "quantity": 2, "price": 500 },
  { "product_id": "102", "quantity": 1, "price": 500.75 }
],
"billingStreetAddress1": "456 Elm Street",
"billingStreetAddress2": "Suite 202",
"billingCity": "Springfield",
"billingState": "Illinois",
"billingPostalCode": "56789",
"companyOrderDate": "2024-11-15",
"companyOrderDueDate": "2024-11-20",
"companyOrderDeliveryStatus": "Pending",
"companyOrderPaymentReference": "PAY-123456",
"companyOrderStatus": "Processing",
"companyOrderDescription": "Order description details",
"company_order_other_attributes_1": { "giftWrap": true },
"company_order_other_attributes_2": { "priorityDelivery": false }
}

```

Update an existing order

<https://productbackend.versal.one/api/order/updateOrder/orderId>

```

{
  // Mandatory Fields
  "company_id": VERPL000972024,

  // Optional Fields
  "firstName": "Jane",
  "lastName": "Smith",
  "phoneNumber": "+9876543210",
  "email": "jane.smith@example.com",
  "postalCode": "12345",
  "streetAddress1": "789 Oak Avenue",

```

```
"streetAddress2": "Unit 5C",
"city": "Shelbyville",
"state": "Indiana",
"specialInstructions": "Call upon arrival.",
"totalAmount": 2000.50,
"products": [
  { "product_id": "201", "quantity": 3, "price": 600 },
  { "product_id": "202", "quantity": 2, "price": 400.25 }
],
"billingStreetAddress1": "321 Pine Lane",
"billingStreetAddress2": "Office 101",
"billingCity": "Shelbyville",
"billingState": "Indiana",
"billingPostalCode": "12345",
"companyOrderDate": "2024-11-10",
"companyOrderDueDate": "2024-11-25",
"companyOrderDeliveryStatus": "Shipped",
"companyOrderPaymentReference": "PAY-654321",
"companyOrderStatus": "Shipped",
"companyOrderDescription": "Updated order description",
"company_order_other_attributes_1": { "giftWrap": true },
"company_order_other_attributes_3": { "priorityDelivery": true }
}
```

Update Order Status via API

To update an existing order in vCX you can call the order update API as below

```
https://productbackend.versal.one/api/order/updateOrder/<orderId>
```

```
{
  // Mandatory Fields cannot be changed
  "company_id": VERPL000972024,

  // Optional Fields
  "firstName": "Jane",
  "lastName": "Smith",
  "phoneNumber": "+9876543210",
  "email": "jane.smith@example.com",
  "postalCode": "12345",
  "streetAddress1": "789 Oak Avenue",
  "streetAddress2": "Unit 5C",
  "city": "Shelbyville",
  "state": "Indiana",
  "specialInstructions": "Call upon arrival.",
  "totalAmount": 2000.50,
  "products": [
    { "product_id": "201", "quantity": 3, "price": 600 },
    { "product_id": "202", "quantity": 2, "price": 400.25 }
  ],
  "billingStreetAddress1": "321 Pine Lane",
  "billingStreetAddress2": "Office 101",
  "billingCity": "Shelbyville",
  "billingState": "Indiana",
  "billingPostalCode": "12345",
  "companyOrderDate": "2024-11-10",
  "companyOrderDueDate": "2024-11-25",
  "companyOrderDeliveryStatus": "Shipped",
  "companyOrderPaymentReference": "PAY-654321",
  "companyOrderStatus": "Shipped",
}
```

```
"companyOrderDescription": "Updated order description",  
"company_order_other_attributes_1": { "giftWrap": true },  
"company_order_other_attributes_3": { "priorityDelivery": true }  
}
```

Getting updates from vCX

If you want to receive updates into your system, when an order is changed in vCX, this could be via the API or manually, you can incorporate your webhooks to receive this information

Getting updates from vCX

Adding webhook for order updates

Webhook to Get Order From VCX

URL

Enter webhook URL

Payload

Add Order Payload

data: {
 company_id:
 req.body.company_id,

Update Order Payload

data: {
 // Fields to update
 // For example, updating the

Update Status Payload

data: {
 // Use one of these based on the
 intended update:

Adding your webhook to his page will enable subscribing to events for order