FRAKTJAKT

Fraktjakt API

Version 3.1.1 - EN 2018-09-22

Table of contents

Welcome!

Fraktjakt is a service for comparing, purchasing and managing shipping services over the Internet. It is available both as a website (https://www.fraktjakt.se) and as an API (*Application Programming Interface*) that can be called by other services on the Internet. This document describes how you can use the ***Fraktjakt API*** to integrate Fraktjakt with other systems and services, such as webshops, business systems, on-line marketplaces, etc.

The basic philosophy of the Fraktjakt API is that it should be as simple as possible to integrate any system with Fraktjakt. Therefore, we try to keep details such as configurations and settings separate from the API; these tasks are performed manually using the Fraktjakt website instead.

In order to use the Fraktjakt API, you need to have some knowledge of *HTML* and *XML*.

The current version of the API is solely XML-based. We may add interfaces based on Web Services or SOAP if we see that there is a demand for this.

Before you integrate Fraktjakt.se with your webshop, please visit https://www.fraktjakt.se/services/download/ to see if we have an existing shipping extension for your webshop platform. If so, you can download the extension and its documentation instead of doing an integration from scratch.

During your integration work you may need to change some settings on your registered Fraktjakt account. Instructions on how to do this (currently only in Swedish) can be found in the Fraktjakt webshop manual for integrators:
https://www.fraktjakt.se/downloads/fraktjakt_manual_webbutik.pdf

The latest version of the document you are reading right now can be found at:

https://www.fraktjakt.se/downloads/Fraktjakt_API_en.pdf

If you encounter any problems while trying to integrate your system with Fraktjakt, we will be glad to help out. We appreciate any questions and comments that you may have about our product and about this documentation.

Use the following link to contact us: https://www.fraktjakt.se/om_fraktjakt/kontakt

Versions

Important changes made in this document

Version	Date	Change and additions
3.1.1	20180922	The “Change and additions” text describing the new elements <status> and <access_link> added in version 3.1 have been corrected (Requery and Track & Trace were added where missing).
3.1	20180915	New URL's for TEST API and PROD API environments (using https). All displayed Fraktjakt URL's and clickable links now have https:// prefix. Added section "Register a webshop that is linked to your existing company" to cover those cases where the user already has a company registered at Fraktjakt. A new element <status> containing the API call status is now included in the response to the Query, Requery, Track & Trace, Order & Shipment API calls. A new element <access_link> containing a link for managing a shipment is now included in the response to the Query, Requery, Order & Shipment API calls. A new element <amount> containing the cost of the shipment is now included in the response to the Order API calls. Notable additions or changes since the last version of this document are indicated with a yellow background.
3.0	20180611	Synchronized with the Swedish version of this document dated 20180427. All text that has been added since the last version of this document is indicated by a green background. A new API has been added – The Shipment API. Two sections have been added to describe XML element blocks that are common to more than one API - specifically the consignor and address elements and their subelements. A section about integration scenarios has been added. New webshop settings that have been introduced in Fraktjakt are described. The relationships between accounts, companies and webshops have been simplified. Descriptions of existing functions have been clarified and improved. New “redirect” parameters have been introduced in order to receive API responses from Fraktjakt in different formats. A shipment can now always be accessed/managed via a single uniform URL - https://www.fraktjakt.se/shipments/show/SHIPMENT_ID&access_code=A

		CCESS_CODE Changes related to adresses: • city_name is no longer required for countries where postal code is required. Fraktjakt determines the city name based on the postal code. • country_subdivision has been eliminated. It shall not be included in any API calls. • Recipient addresses may now include in country codes. Fraktjakt now supports customised notifications. export_reason is now required for both export and import shipments. Improved language support. Language can now be specified for notifications to receivers of shipments. Appendix 1 and 2 added.
2.91	20150225	New non-obligatory elements in Consignor field. **system_name** – name of the platform sending the API call. It can be the name of your webshop platform, i.e. ”Magento”, Prestashop”, ”WooCommerce”, etc. **system_version** – version number of the system mentioned in the ”system_name” field above. **module_version** – version number of the Fraktjakt shipping module in use. **api_version** – version number of the Frakjakt API in use (should be the same number as this documents version number!).
2.9	20140616	Clarification of the API's functionality: 1) a parcels length, width and height must be provided (Obligatory: Yes) if the <parcel> element is included in the call to Order API call type 2 [see Order API Call section]. 2) to create an order from a previous query call, a parcels length, width and height must be provided in the original query call [see Query API Call section].
2.8	20140603	New element <*shipper_info*> in the Query and Requery API to switch on/off additional information about the shipper (in the Api response) for each returned *shipping_product*. New element <*name*> under <*shipping_product*>, where the shipping product name is returned without also containing the shipper name. Corrected name of element (from `time_guarantie` to `time_guarantee`).
2.7	20140307	The elements <*commodities*> and <*parcels*> are no longer required when calling Order API type 2 to order *Fraktbag* products (*shipping_product_id* = 137 – 141).

2.6	20140305	*Reason for export* must be entered into the Order API for international shipments. Added some additional XML examples for Query and Order API calls.
2.5	20140301	Support for *Fraktjakt+* and *Eget fraktavtal (individual shipping contracts)* subscriptions.
2.4	20140225	Fixed bug in multiple multiple shipments handling. Corrected section about *”Error messages from the API”*. Added more error messages and solutions. Address handling explained under *Order API*.
2.3	20140224	Added functionality in the *Order API*, to enable creation of an order without creating a shipment through *Query API* first. New Fraktjakt API test page Track & Trace API (trace_xml) now requires authentication (include Consignor ID & Key). Fixed bug in MD5 handling, to respond with a faster cached reply, when you have entered the same *Query API* call (within 24 hours).
2.2	20140218	Minor update. New interaction diagram. Layout and language corrections.
2.1	20140217	Minor update. Fixed translation, examples, hyperlinks and new logo.
2.0	20130819	Updated API documentation to the new Fraktjakt v2.0
1.16	20110220	New link how to track parcels not from Fraktjakt. Anonymous sender is now possible on the shipping-documents.
1.15	20101228	*Trace-API* to keep track on your shipments
1.14	20101118	Better explanation for how to add commodities to a package to get the right measurements.
1.13	20101021	Examplification on how to build a parcel
1.12	20101020	More info about how to make an integration.
1.11	20100921	webshops API (Create an account)

Why integrate with Fraktjakt?

Because it's completely free to use, and it will save you time and money!

If you have a webshop, Fraktjakt gives you **simplified shipping management** and **discounted shipping prices**. Your customers will be able to choose the quickest, cheapest or most convenient shipping alternative when they purchase products on your site.

Fraktjakt eliminates the need to procure your own shipping contracts, since we have already negotiated volume based shipping contracts for both domestic and international shipping. You get access to highly discounted prices when you use Fraktjakt through an API integration.

Furthermore, Fraktjakt also simplifies management of your shipping, so you can focus more on selling your products. When you integrate Fraktjakt's order handling system with your application's sales functions, you are essentially plugging in your system to a variety of shipping providers. Fraktjakt manages all of your communication with the shippers. It also generates the necessary shipping documents, etc.

Configuration Possibilities with Fraktjakt

- Selection of which shipping services your customers may choose when they make purchases in your webshop;
- Selection of whether to sort the shipping alternatives by **time** or **price** when they are presented to your customers;
- Selection of whether shipping time will be displayed as **Number of days** or **Arrival time**;
- Selection of **Free shipping** when a customer's purchase exceeds a specific amount;
- Setting a price for picking up a product in person instead of having it shipped;
- Definition of optional standard box dimensions, if you want Fraktjakt to use these;
- Selection of which days shippers can pick up products for delivery;
- Definition of pick-up times, if applicable;
- Custom track-and-trace page (with your logo, etc.);
- Custom shipping dispatch notification e-mails to your customers;
- Control over which categories of e-mails you receive from Fraktjakt.

Useful functions

- Fraktjakt can handle all products in the customer's shopping cart, regardless of whether they are sent as separate parcels or if multiple products are packaged together in a standard box.
- Fraktjakt allows you to define the dimensions of individual products. This provides more exact price calculations.
- Fraktjakt can determine which products in your customer's shopping cart will fit into your standard boxes and base the shipping prices on your standard box size. This is normally cheaper than basing the shipping prices on the dimensions of your individual products.
- Fraktjakt estimates delivery times based on your settings, i.e. which days shippers can come and pick up products for delivery, along with any specific pick-up times that you may have defined.
- Your customers only see the shipping alternatives that you want them to be able to choose.
- When your customers choose delivery that is not door-to-door, they are shown the nearest pick-up location and how far it is from their address to that location.

- Fraktjakt provides links to maps that help your customers locate pick-up locations.
- Fraktjakt prepares booking and pick-up of one or more shipments.
- Frakt creates shipping documents for both domestic and international shipments, to be printed out on your own printer.

Server environments

Fraktjakt has two separate server environments that you can use with your API integration.

TEST API – the test environment. URL: https://testapi.fraktjakt.se

The test environment is a sandbox for use under development and testing. Here you can send as many API calls as you want to create test shipments without incurring any actual costs. All other functionality is the same as in the live production environment except that shipping documents are created with watermarks (and are therefore not valid) and the payment process is a simulation (no real payment charges).

NOTE: The test environment may offer shipping alternatives that aren't actually purchasable via Fraktjakt. Conversely, some shipping alternatives that are purchasable via Fraktjakt may not be offered in the test environment.

NOTE: Remember that the transactions that are performed in the test environment are not necessarily saved very long. We occasionally may purge the test server data to keep it efficient.

PROD API – the production environment. URL: https://api.fraktjakt.se

The production environment shows shipping services that can actually be purchased via Fraktjakt.

This is where you send your API calls when you have finished testing and are ready to purchase shipping services for real through Fraktjakt.

This environment is optimized to provide more efficient searches and better stability and security for your webshop.

NOTE: Avoid sending API calls to https://www.fraktjakt.se. This could seriously slow down your application, provide unpredictable results and make it more difficult for us to troubleshoot any API-related problems you encounter.

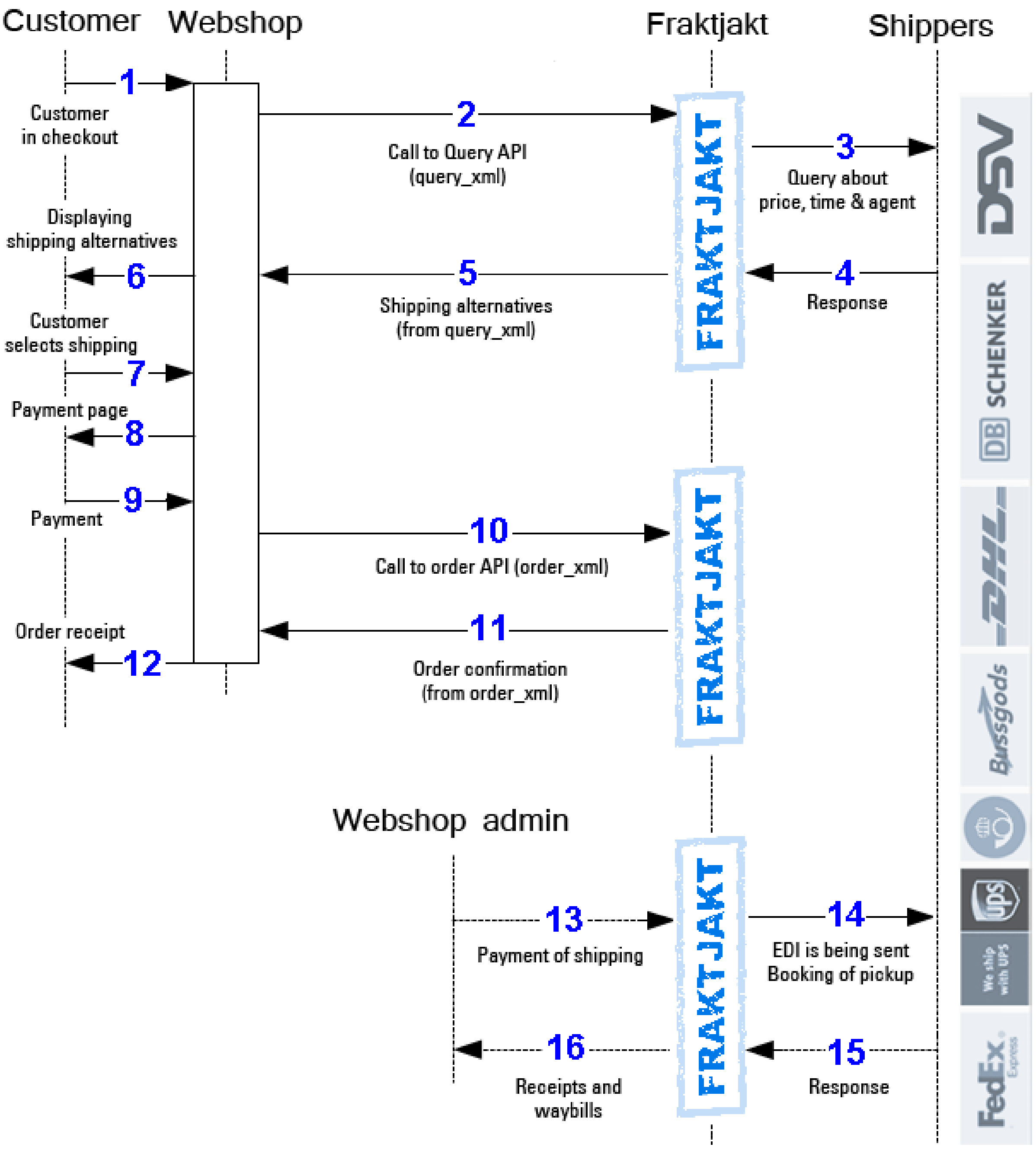
NOTE: The https://www.fraktjakt.se URL should only be used from your web browser, when you are managing your shipments, printing out your shipping documents, configuring your Fraktjakt account, etc.

Interaction Diagram

The following diagram depicts the flow of interaction between a webshop customer, the webshop, and Fraktjakt.

The details that are shown of the interaction between the webshop customer and the webshop are just an example. The exact details will vary, depending on the workings of each individual webshop.

Note that the webshop manager needs to manually log on to the Fraktjakt website in order to pay for the shipping services that the webshop's customers have ordered. Multiple shipping orders can be paid for and handled simultaneously using Fraktjakt's transport administration interface.


Customer
Webshop
Fraktjakt
Shippers
1
Customer in checkout
2
Call to Query API (query_xml)
3
Query about price, time & agent
4
Response
5
Shipping alternatives (from query_xml)
Displaying shipping alternatives
6
Customer selects shipping
7
Payment page
8
9
Payment
10
Call to order API (order_xml)
11
Order confirmation (from order_xml)
Order receipt
12
FRAKTJAKT
DSV
DB SCHENKER
DHL
Bussgods
ups
We ship with UPS
FedEx Express
Webshop admin
13
Payment of shipping
14
EDI is being sent Booking of pickup
15
Response
16
Receipts and waybills

Integration Scenarios

Listed below are some of the most common scenarios when using the API's.

1. Several shipping services are presented to the customer

Webshops that want to let their customers choose the shipping service can use the *Query API* in combination with *Order API, type 1*. The customers choice may be based on the price or maybe instead on the proximity to the agent location where they will pick up their package.

- The webshop performs a price query using *Query API*.
- The query results are displayed as alternative shipping services in the webshop shopping cart or checkout.
- The customer chooses one of the presented shipping services.
- The webshop creates a shipment order in Fraktjakt by sending the customer selected shipping service in a call to *Order API, type 1*.
- The webshop administrator manages the new shipment order in Fraktjakt via this link: https://www.fraktjakt.se/shipments/show/SHIPMENT_ID?access_code=ACCESS_CODE

2. One shipping service is presented to the customer

The above query can be narrowed down to just one specific shipping service in the case that the webshop does not want the customer to decide on the shipping service, since it wants to always use just one specific shipping service.

- Retrieve the price of only one shipping service using *Query API*, see "*Example 3 - Search for a specific shipping service (enter 'shipping_product_id')*" for details.
- The query result is displayed as the only shipping service in the webshop shopping cart or checkout.
- The webshop creates a shipment order in Fraktjakt by sending the shipping service in a call to *Order API, type 1*.
- The webshop administrator manages the new shipment order in Fraktjakt via this link: https://www.fraktjakt.se/shipments/show/SHIPMENT_ID?access_code=ACCESS_CODE

3. Fixed shipping price, using webshop's pre-determined shipping service

This scenario illustrates a webshop that prefers to offer a fixed shipping price to its customers and wants to always use a specific pre-chosen shipping service to fulfill all of its order deliveries. This can be facilitated by using the *Order API, type 2* which can directly create a new shipment order in Fraktjakt without necessitating a price query in advance.

- Find the available shipping services at Fraktjakt using this listing: https://www.fraktjakt.se/shipping_products/xml_list
- The webshop creates a shipment order in Fraktjakt by sending the shipping service in a call to *Order API, type 2*.

- The webshop further manages the shipment order via the link: https://www.fraktjakt.se/shipments/show/SHIPMENT_ID?access_code=ACCESS_CODE

4. Fixed shipping price, webshop chooses shipping service later on

A webshop that wants to offer a fixed shipping price to its customers and wants to easily transfer the order details into a new shipment order in Fraktjakt. The webshop later on manages the shipment order in Fraktjakt, choosing the appropriate shipping service itself.

- Send all the customer information, including delivery address, and the purchased item details to Fraktjakt using the *Shipment API*.
- The webshop further manages the shipment order via the link: https://www.fraktjakt.se/shipments/show/SHIPMENT_ID?access_code=ACCESS_CODE

5. Webshop portal searches and orders shipping for a member webshop

This is useful when a webshop portal integrates multiple webshop customers with Fraktjakt.

- The webshop portal does a price query with *Query API,* possibly searching for individual shipping services by adding a shipping_product_id.
- A customer chooses one of the presented alternatives.
- The webshop portal submits the selected shipment to Fraktjakt using *Order API, type 1* (alternatively 2), with the integration credentials of its member webshop using the referred_consignor-element.
- Fraktjakt returns a link by which the shipping purchase can be completed by the member webshop.

6. Show what's happening with a shipment after it's been purchased in Fraktjakt

- Fraktjakt has many settings that allow webshops to create a customized page for customers to track their shipments in a user-friendly manner.
- You can also create your completely own parcel tracking page with the *Track & Trace API* på sidan 55.

Register a company with a webshop integration

Before your system can communicate with Fraktjakt, you must have a *Consignor ID* and a *Consignor key.*

NOTE! The instructions below show links to both the TEST API and PROD API environments. Please use the TEST API links during integration work (development and testing). Only use the PROD API links once you have everything working satisfactorily and you are ready for production.

1. If you already have a Fraktjakt user account, log in to it now.

   TEST API: https://testapi.fraktjakt.se/account/login (Recommended for initial testing!)

   PROD API: https://api.fraktjakt.se/account/login

2. If you already have a company registered with Fraktjakt, go to the next section "*Register an online store that is linked to your company*". Otherwise, go to the "**Register a Company**" page:

   TEST API: https://testapi.fraktjakt.se/account/register (Recommended for initial testing!)

   PROD API: https://api.fraktjakt.se/account/register

3. Fill out the entire form and indicate that you want a **Webshop integration** (important). Then press the [Register] button at the bottom of the page.

4. Enter the activation code in the appropriate input field. The code should be sent to the email address you provided during registration.

5. Now you should be logged in! (or, still logged in if you started with step 1). Your user name should appear in the web browsers upper right corner.

6. Change the active account to the new webshop, in the top right corner of the browser. It appears there as a menu choice under both your private user account and your registered company.

7. Then go to the "**Settings**" page of your webshop:

   TEST API: https://testapi.fraktjakt.se/webshops/change (Recommended for initial testing!)

   PROD API: https://api.fraktjakt.se/webshops/change

8. Here you will find the **Consignor ID** and **Consignor Key** that you will soon need to submit API calls to Fraktjakt. You can also change your settings that affect the API calls and responses.

Register a webshop that is linked to your existing company

Before your system can communicate with Fraktjakt, you must have a *Consignor ID* and a *Consignor key.*

NOTE! The instructions below show links to both the TEST API and PROD API environments. Please use the TEST API links during integration work (development and testing). Only use the PROD API links once you have everything working satisfactorily and you are ready for production.

1. If you already have a Fraktjakt user account, log in to it now.

   TEST API: https://testapi.fraktjakt.se/account/login (Recommended for initial testing!)

   PROD API: https://api.fraktjakt.se/account/login

2. If you already have a company registered with Fraktjakt, change the active account to your company. It appears in the browser's upper right corner as a menu selection under your private user account.

3. Then go to the "**Settings**" page of your company:

   TEST API: https://testapi.fraktjakt.se/enter_shipper/verify (Recommended for initial testing!)

   PROD API: https://api.fraktjakt.se/enter_shipper/verify

4. Under the *Administration* section, click on *Webshops*.

5. Then click the button "+ **Register webshop**".

6. Enter which webshop system you are using, and then click the "**Save**" button.

7. Then go to the "**Settings**" page of your webshop by clicking *Webshops* again in the *Administration* section, and then clicking on the name of the new webshop you have just registered.

8. Here you will find the **Consignor ID** and **Consignor Key** that you will soon need to submit API calls to Fraktjakt. You can also change your settings that affect the API calls and responses.

Important:

Consignor ID and Consignor key are never the same in the test environment (TEST API) and production environment (PROD API).

Test is a sandbox environment and is in no way connected to production.

It's not possible to purchase shipping services for real using the test environment.

Generall information about the XML in the API calls

The API supports both *UTF-8* and *ISO-8859-1* encoding, but we recommend *UTF-8* for all communication with Fraktjakt. If you are using the PHP programming language, it is best to choose *ISO-8859-1*, since *UTF-8 t*ends to cause problems in PHP programs.

The calls that you send to Fraktjakt over the API are sent as HTTP requests.

All decimals that appear in numeric values must be decimal points – not commas. Example: ”3.14”

In this documentation we use indentation to show that an XML element is contained by the element on the previous line of XML.

The XML is sent in as a parameter named xml in the URL to Fraktjakt. It has to be URL-encoded. https://en.wikipedia.org/wiki/Query_string#URL_encoding

Common to all API calls

All calls to Freight Charge API have some data that will always be sent, regardless of which API you call.

1. The ”*xml*” parameter must be included in all calls made over the API. This parameter defines the XML data that applies to the call. The XML data should only contain standard XML elements written in lower-case. If the data that you need to send over the API is in an XML-file, you can simply remove all of the line breaks from the file and convert it to a long string of text. Then you use this text as the value of the ”xml” parameter in your call to Fraktjakt.

2. There is an optional parameter named ”*md5_checksum*”. This is the MD5 checksum for the XML data that is defined by the *”xml”* parameter (See https://en.wikipedia.org/wiki/Md5). If you define the *”md5_checksum”* parameter in a call to Fraktjakt, the checksum will be used to verify the contents of the *”xml”* parameter. The *”md5_checksum”* parameter is optional, but including it will help to speed up the caching of API calls.

3. Remember to URL-encode the XML-string according to this: https://en.wikipedia.org/wiki/Query_string#URL_encoding

Consignor element

All calls to the API **must** include the consignor element. This element tells Fraktjakt who is making the call (through the Consignor ID and Consignor Key combination) and provides some other information that is needed to process the call. Without a properly supplied consignor element a call will not succeed.

<consignor>	Information about the consignor, i.e. the webshop or other entity that is sending the order.
<id>**YOUR_CONSIGNOR_ID**</id>	The consignor's customer ID in Fraktjakt. Data type: INT Required: Yes Your webshops id in Fraktjakt. Found in the installation tab for the webshop in Fraktjakt.
<key>**YOUR_CONSIGNOR_KEY**</key>	The consignor's login key in Fraktjakt. Data type: STRING Max chars: 64 Required: Yes Found in the installation tab for the webshop in Fraktjakt.
<currency>**SEK**</currency>	Currency in ISO 4217 notation. Data type: STRING Max chars: 3 Required: No Currency for the purchase in ISO 4217. Only compatible with 'SEK', (Swedish currency). Optional for now.
<language>[**sv | en**]</language>	Language in ISO 639-1 notation (6) Data type: STRING Max chars: 2 Required: No Default: sv See Appendix 1 page 76
<encoding>**[UTF-8 | ISO-8859-1]**</encoding>	The encoding that applies to text in the returned XML data. Data type: STRING Required: No Default: UTF-8 UTF-8 is the default and recommended encoding. ISO-8859-1 is also compatible.
<system_name>Prestashop</system_name>	Name of system calling the API. Data type: STRING Max chars: 32

	Required: No Woocommerce, Mangento Prestashop etc.. This is to make troubleshooting and support easier
`<system_version>1.2.3</system_version>`	Version number of system named in `<system_name>` element. Data type: STRING Max chars: 16 Required: No This is to make troubleshooting and support easier
`<module_version>1.5.0</module_version>`	Version number of the Fraktjakt shipping module in use. Data type: STRING Max chars: 16 Required: No
`<api_version>3.1.0</api_version>`	Version number of the API integrationen in use. Data type: STRING Max chars: 16 Required: No Should be the same version as this document!
`</consignor>`	

Address elements

In order for a shipment to be created, address elements must be provided, such as address_to, addreas_from, etc.

The element content (between the element tags) must be as follows:

`<address_*> (<address_from>, <address_to>, etc.)`	Address type Datatype: STRING Max chars: 25 Obligatory: Ja Replace the "*" in "<address_*", at left, with the address type, e.g. "to", "from", ...
`<street_address_1>`**`Gjuterig 9`**`</street_address_1>`	Gatuadress (line 1) Datatyp: STRING Max chars: 255 Obligatory: Yes If the shipment is to a 'C/O' address should be in this element.
`<street_address_2>`**`Vån 7 C`**`</street_address_2>`	Gatuadress (line 2) Datatype: STRING Max chars: 255 Obligatory: No Supplementary/further shippinginformation can be given within parentheses in this element. Ex: (Has unloading agreements) (Leave at the loading platform)
`<postal_code>`**`55318`**`</postal_code>`	Postalcode Datatype: STRING Max chars: 16 Obligatory: Yes – except for countries not using postal codes, like Gambia, see Appendix 2, Postal code is also the primary way of finding the name of the city to which the shipment is going. The <city_name> element will only be used unclear cases.
`<city_name>`**`Jönköping`**`</city_name>`	City Datatype: STRING Max chars: 32 Obligatory: Not if the <postal_code> line contains a unique postal code for that city.
`<residential>`**`[0	1]`**`</residential>`	Specifies whether the address is residential or not (private/residential or commercial) Datatype: BOOLEAN Obligatory: No Default = 1 (residential)
`<country_code>`**`SE`**`</country_code>`	Country - See Appendix 2 Datatype: STRING Max chars: 2 Obligatory: Yes

`<language>`**`sv`**`</language>`	Receiver language, see Appendix 1 page 76. Language code: ISO 639-1 Datatyp: STRING Max chars: 2 Can **ONLY** be given for <address_to>-elements. Specifies which language Fraktjakt should use in email and communication with the receiver.
`</address_*> (</address_from>, </address_to>, etc.)`	

Fraktjakt Query API

The API calls that are used for sending queries about shipping services to Fraktjakt are collectively called the Fraktjakt Query API. In this document we will simply use the term ”Query API”.

Your system can use the Query API to retrieve information about the best, cheapest and fastest shipping alternatives. This is the same type of information that web users find when they perform searches at https://www.fraktjakt.se/; however, the Query API is faster.

XML for Query API calls

In order to search for shipping services, your system must send a definition of the required shipping services to the Query API. The definition that is sent may comprise one or more parcels and the address that they need to be shipped to. The definition must only have one sender address and one receiver address, regardless of how many parcels are included.

Each parcel that is included in a query definition must be accompanied by the parcel's weight. Width, length, and height dimensions are optional, but it is recommended that these dimensions are included in order to provide more accurate query results. If the volume of each parcel is not included in the query definition, Fraktjakt may apply a default volume. You can define this default volume on your webshop configuration page at Fraktjakt.

If the shipping price is distance-based, the shipping address that you entered on your webshop configuration page at Fraktjakt will be used by default.

The Query API uses the following URI:s:

TEST API environment:	https://testapi.fraktjakt.se/fraktjakt/query_xml
PROD API environment:	https://api.fraktjakt.se/fraktjakt/query_xml

If you want to get the price a specific shipping service the <shipping_product_id> element can be used. You can retrieve the ID-number from an earlier search in the <id> element under the <shipping_product> element or via https://www.fraktjakt.se/shipping_products/xml_list.

You can configure Fraktjakt to send <arrival_time> as the expected shipping time in days or as the expected time of arrival. This setting is done on your configuration page at Fraktjakt.se. The arrival time is calculated based on the shipping product's normal shipping time, the available days for pick-up that you have defined in your settings, and the shipper's pick-up schedule.

Note that if you send a query to the Query API that is an exact duplicate of a previous query, the Query API will return the cached results from the first query.

When the response is more important than precision.

You can also do a quick search that we cache locally. It will provide an estimate shipping price quickly when your customers do not need the exact final price. This is also useful when you don't have the receiver address yet.

To do this you put the same address in the receiver address field as in the sender field. This will tell

Fraktjakt to make a quick search.

NOTE! Do not create an order from this search. You need to do a new search when you want to create the order and buy the shipment. Otherwise the incorrect address will be sent to the delivery company. Then it could be suitable to set the search for a specific shipping_product_id.

XML element tag and example of contents	**Description**
<?xml version="1.0" encoding="UTF-8"?>	XML header
- <shipment>	This is the document element, i.e. the root element of the XML data that describes a shipment that is being searched for.
- <consignor>	Information about who's making the call.
See Consignor-element on page 17	
</consignor>	
- <value>**10**</value>	The cummulative value of all items in the shipment. Data type: FLOAT Required: No
- <express>**[0 | 1]**</express>	Flag for limiting the search to only find express shipping alternatives. Data type: BOOLEAN Required: No Default = 0 (Search for both express and other types of shipping services.)
- <pickup>**[0 | 1]**</pickup>	Flag for limiting the search to only find shipping alternatives with pickup service. Data type: BOOLEAN Required: No Default = 0 (Do not limit the search.)
- <dropoff>**[0 | 1]**</dropoff>	Flag for limiting the search to only find shipping alternatives with dropoff service. Data type: BOOLEAN Required: No Default = 0 (Do not limit the search.)
- <green>**[0 | 1]**</green>	Flag for limiting the search to only find shipping alternatives that have some type of environmental certification. Data type: BOOLEAN Required: No Default = 0 (Do not limit the search.)
- <quality>**[0 | 1]**</quality>	Flag for limiting the search to only find shipping alternatives that have some type of quality certification. Data type: BOOLEAN Required: No Default = 0 (Do not limit the search.)
- <time_guarantee>**[0 | 1]**</time_guarantee>	Flag for limiting the search to only find shipping alternatives that have guaranteed delivery time. Data type: BOOLEAN Required: No Default = 0 (Do not limit the search.)
- <cold>**[0 | 1]**</cold>	Flag to indicate whether the shipment

	contains items that require refrigeration. (3) Data type: BOOLEAN Required: No Default = 0 (refrigeration not required)
- <frozen>**[0 | 1]**</frozen>	Flag to indicate whether the shipment contains items that require frozen transport. (3) Data type: BOOLEAN Required: No Default = 0 (frozen transport not required)
- <shipping_product_id>**30**</shipping_product_id>	Include this element if you only want to find the price for a specific shipping product. The ID will normally be the value found in the <id> element of a shipping product that was returned by a prior query. [NB: This element can be useful if you want to fill in a price using AJAX.] Data type: INT Required: No
- <no_agents>**[0 | 1]**</no_agents>	If you want to know the nearest agent or not. If you select to not show them you will receive a faster reply, but give less service to your customers. Data type: BOOLEAN Required: No Default = 0 (show agents)
- <no_prices>**[0 | 1]**</no_prices>	No prices are shown! Gives a much faster result, but are only showing if a service is available or not to a given address. Can be combined with <no_agents>. Use it if only availability is of interest. Data type: BOOLEAN Required: No Default = 0 (prices are shown)
- <agents_in>**[0 | 1]**</agents_in>	If you want to show the nearest agent for the sender or not. If you select to show it, the reply will be much slower. Data type: BOOLEAN Required: No Default = 0 (don't show agents)
- <shipper_info>**[0 | 1]**</shipper_info>	If you want extra information about the shippers. The reply will then contain the additional element <shipper> with its sub element's *id, name* and *logo_url*. Datatyp: BOOLEAN Required: No Default = 0 (no additional info)
- <parcels>	The parcels that the shipment comprises. At least one parcel must be included, and we recommend that no more than 10 parcels should be included in a single shipment. MinOccur = 1 **in call type 2.** **Exckluded in call type 2 if ordering a *Fraktbag*** (shipping_product_id = 137-141).

	Always excluded in call type 1
- <parcel>	Data pertaining to a parcel.
<weight>**3.2**</weight>	The parcel's weight in kg. Data type: FLOAT Required: Yes
<length>**35**</length>	The parcel's length in cm. The length is the longest side of the parcel. Data type: FLOAT Required: No
<width>**23.5**</width>	The parcel's width in cm. The width is the second longest side of the parcel. Data type: FLOAT Required: No
<height>**15**</height>	The parcel's height in cm. Data type: FLOAT Required: No
</parcel>	
</parcels>	
- <address_to>	Note that <language> can be used.
Se address-element on page ?	
</address_to>	
- <address_from>	Sender address (If this element is missing the address entered in Fraktjakt will be used. Obligatory: No(see note nr.1)
See Address-element på on page ?	
</address_from>	
</shipment>	

Notes:

1. <*address_from*> normally does not need to be included, since the shipping address that you have entered on your configuration page at Fraktjakt.se should apply to all of the shipments that you order via Fraktjakt. See step 5 in the section of this document titled *Creation of a webshop account at Fraktjakt.se* (above).

   The Query API supports addresses anywhere in on the globe; however, Fraktjakt does not handle shipments from countries other than Sweden at the time of this writing.

2. If the elements, <*cold*> and <*frozen*>, are omitted from your XML, or if they do not contain the value ”1” or ”true”, no refrigerated or frozen shipping alternatives will be included in the search results that the Query API returns. Furthermore, refrigerated and frozen shipping alternatives are only shown when these elements contain ”1” or ”true”. It is thus not possible to get a search result that includes both refigerated/frozen and non-refrigerated shipping alternatives in the same call to the Query API.

3. The origin or destination address must be in Sweden.

4. Currently, only 'sv' is supported. The standard is described at https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

Example 1 - Regular shipping query

A regular search delivers a response with one or more shipping alternatives.

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <value>199.50</value>
  <shipper_info>1</shipper_info>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <no_agents>1</no_agents>
  <parcels>
    <parcel>
      <weight>2.8</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
  </parcels>
  <address_to>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
    <language>sv</language>
  </address_to>
</shipment>
```

Example 2 - Regular shipping query with multiple packages

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <value>199.50</value>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <no_agents>1</no_agents>
  <parcels>
    <parcel>
      <weight>1.2</weight>
      <length>20</length>
      <width>15</width>
      <height>10</height>
    </parcel>
    <parcel>
      <weight>2.8</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
    <parcel>
      <weight>7</weight>
      <length>40</length>
      <width>30</width>
      <height>20</height>
    </parcel>
  </parcels>
  <address_to>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
    <language>sv</language>
  </address_to>
</shipment>
```

Example 3 - Search for a specific shipping service (enter 'shipping_product_id')

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <value>199.50</value>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <no_agents>1</no_agents>
  <shipping_product_id>25</shipping_product_id>
  <parcels>
    <parcel>
      <weight>2.8</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
  </parcels>
  <address_to>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
    <language>sv</language>
  </address_to>
</shipment>
```

Example 4 - Quick Search

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <value>199.50</value>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <no_agents>1</no_agents>
  <shipping_product_id>25</shipping_product_id>
  <parcels>
    <parcel>
      <weight>2.8</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
  </parcels>
  <address_from>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
  </address_from>
  <address_to>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
    <language>sv</language>
  </address_to>
</shipment>
```

Example 5 - Regular FOREIGN shipping query

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">
  <value>199.50</value>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <parcels>
    <parcel>
      <weight>2.8</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
  </parcels>
  <address_to>
    <street_address_1>101 Main Street</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>12345</postal_code>
    <city_name>Schenectady</city_name>
    <residential>1</residential>
    <country_code>US</country_code>
    <language>en</language>
  </address_to>
</shipment>
```

XML for Query API responses

XML element tag and example of contents	Description
<?xml version="1.0" encoding="UTF-8"?>	XML header
- <shipment>	This is the document element, i.e. the root element of the XML data that describes a shipment that is being searched for.
<status>**ok**</status>	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
<code>**1**</code>	Error code 0 = OK 1 = Warning 2 = Error
<warning_message>**You need to input all measurements**</warning_message>	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
<error_message></error_message>	Error message. Data type: STRING Max chars: 500 Required: Yes
<currency>**SEK**</currency>	Currency. Data type: STRING Max chars: 3 Required: Yes
<id>**67887**</id>	Fraktjakt's SHIPMENT_ID. This is used in all future references to this particular query result. Data type: INTEGER Required: Yes
<access_code>**ABC12345**</access_code>	A code for accessing and managing the shipment without previously being logged in. (3)
<access_link> https://www.fraktjakt.se/shipments/show/163221?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9 </access_link>	A link to the shipment using the above SHIPMENT_ID and ACCESS_CODE. Data type: STRING Required: No
- <shipping_products>	The query result is presented as an XML array of shipping products, sorted in the order defined on your configuration page at Fraktjakt.se.
- <shipping_product>	An individual shipping product, i.e. a service that matches the search criteria that were sent in the query. See page 61 for the shipping services

	currently possible to get in the response.
<id>**15**</id>	The shipping product's ID in i Fraktjakt. This can be used for purchasing the shipping product. Data type: INTEGER Required: Yes
<description>**Posten AB Postpaket**</description>	Description of the shipping product. This can be presented directly to your customers. Data type: STRING Max chars: 35 Required: Yes
<arrival_time>**One to three days, before 10 am guaranteed**</arrival_time>	The arrival time; either the number of days from now or the expected date/time of arrival (7). Data type: STRING Max chars: 64 Required: Yes, but may be empty
<price>**159.50**</price>	The shipping product's total price. Data type: FLOAT Required: Yes
<tax_class>**25.00**</tax_class>	The VAT percentage that applies to this shipping product. Data type: FLOAT Required: Yes
<agent_info>**Cityterminalen Stockholm apx 2 km in Stockholm**</agent_info>	The shipping agent location that is closest to the receiver's address (does not apply to direct delivery shipping products). Data type: STRING Max chars: 72 Required: Yes, but may be empty
<agent_link> **https://www.fraktjakt.se/agents/search_closest/377483?type=9&shipper=4** </agent_link>	A link to a map that shows where the shipping agent defined in <agent_info> is located. Data type: STRING Max chars: 256 Required: Yes, but may be empty
<agent_in_info>**Fedex Jönköping apx 1 km in Jönköping**</agent_info>	The shipping agent location that is closest to the sender's address (does not apply to shipping products with pickup). Is only sent if agents_in is true. Data type: STRING Max chars: 72 Required: No
<agent_in_link> **https://www.fraktjakt.se/agents/search_closest/377482?type=8&shipper=4** </agent_link>	A link to a map that shows where the shipping agent defined in <agent_in_info> is located. Is only sent if agents_in is true. Data type: STRING Max chars: 256 Required: No
<shipper>	The shipper delivering the shipping product. This element (and its sub elements) is only created if *<shipper_info>* has been sent in the initial API call with the value *"1"*. Required: No

`<id>`**4**`</id>`	The shipper's ID # in Fraktjakt. Data type: INTEGER Required: No
`<name>`**Bussgods**`</name>`	The shipper's name. Data type: STRING Max chars: 35 Required: No

`<logo_url>`
https://www.fraktjakt.se/images/shippers/4.png
`</logo_url>`
	The shipper image logo url. In case you want to present it in the search result.
	Datatype: STRING
	Max chars: 72
	Required: No

`</shipper>`	
`</shipping_product>`	
`</shipping_products>`	
`</shipment>`	

Requery API (cached shipment search)

If you've previously called Query API and got a response containing a shipment_id you can call the Requry API to get a cached response which is much quicker.

Use the following URI to get a cached response from Fraktjakt via the Requery API:

https://testapi.fraktjakt.s/fraktjakt/requery_xml (TEST API – test environment)

https://api.fraktjakt.s/fraktjakt/requery_xml (PROD API – live / production environment)

The XML that you send to the Requery API should include all of the standard fields that were sent to the Query API, plus a shipment_id value that the Query API returned.

You can configure Fraktjakt to send <arrival_time> as the expected shipping time in days or as the expected time of arrival. This setting is done on your configuration page at Fraktjakt.se. The arrival time is calculated based on the shipping product's normal shipping time, the available days for pick-up that you have defined in your settings, and the shipper's pick-up schedule.

XML element tag and exampel of contents	**Description**
`<?xml version="1.0" encoding="UTF-8"?>`	XML header
`- <shipment>`	This is the document element, i.e. the root element of the XML data that describes a shipment that is being searched for.
`- <value>`**10.50**`</value>`	Value of content in the shipment. Datatype: FLOAT Obligatory: No
`- <shipper_info>`**[0 | 1]**`</shipper_info>`	Include this element if you want to receive extra information about the

	shipping providers in the response . The response will then contain a <shipper> element containing id, name and logo_url elements. Datatype: BOOLEAN Obligatory: No Default = 0 (iNo extra info)
- <consignor>	Information about who's making the call.
See Consignor-element on page 17	
</consignor>	
<shipment_id>**67887**</shipment_id>	Fraktjakt's shipment ID. This is used in all future references to this particular query result. Data type: INTEGER Required: Yes
</shipment>	

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<shipment>
  <value>10</value>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <shipment_id>67887</shipment_id>
</shipment>
```

Response

XML element tag and example of contents	**Description**
<?xml version='1.0' encoding='UTF-8'?>	
- <shipment>	This is the document element, i.e. the root element of the XML data that describes a shipment that is being searched for.
<status>**ok**</status>	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
<code>**1**</code>	Error code 0 = OK 1 = Warning

	2 = Error
<warning_message>**Du behöver ange alla mått**</warning_message>	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
<error_message></error_message>	Error message. Data type: STRING Max chars: 500 Required: Yes
<currency>**SEK**</currency>	Currency. Data type: STRING Max chars: 3 Required: Yes
<id>**67887**</id>	Fraktjakt's SHIPMENT_ID. This is used in all future references to this particular query result. Data type: INTEGER Required: Yes
<access_code>**ABC12345**</access_code>	A code for accessing and managing the shipment without previously being logged in. (3)
<access_link> https://www.fraktjakt.se/shipments/show/163221?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9 </access_link>	A link to the shipment using the above SHIPMENT_ID and ACCESS_CODE. Data type: STRING Required: No
- <shipping_products>	The query result is presented as an XML array of shipping products, sorted in the order defined on your configuration page at Fraktjakt.se.
- <shipping_product>	An individual shipping product, i.e. a service that matches the search criteria that were sent in the query. See page 63 for the shipping services currently possible to get in the response.
<id>**15**</id>	The shipping product's ID in i Fraktjakt. This can be used for purchasing the shipping product. Data type: INTEGER Required: Yes
<name>**Privat**</name>	The name of the service. Can be presented directly to you customers. Datatype: STRING Max chars: 35 Obligatory: **Ja**
<description>**Bussgods - Privat**</description>	Description of the shipping product. This can be presented directly to your customers. Data type: STRING Max chars: 35 Required: Yes
<arrival_time>**1-2 dagar**</arrival_time>	The arrival time; either the number of days from now or the expected date/time of arrival (7). Data type: STRING Max chars: 64

	Required: Yes, but may be empty
<price>**159.50**</price>	The shipping product's total price. Data type: FLOAT Required: Yes
<tax_class>**25.00**</tax_class>	The VAT percentage that applies to this shipping product. Data type: FLOAT Required: Yes
<agent_info>**Cityterminalen Stockholm ca 2 km i Stockholm**</agent_info>	The shipping agent location that is closest to the receiver's address (does not apply to direct delivery shipping products). Data type: STRING Max chars: 72 Required: Yes, but may be empty
<agent_link> **https://www.fraktjakt.se/agents/search_closest/377483?type=9&shipper=4** </agent_link>	A link to a map that shows where the shipping agent defined in <agent_info> is located. Data type: STRING Max chars: 256 Required: Yes, but may be empty
<agent_in_info>**Jönköping Bussgods ca 1 km i Jönköping**</agent_info>	The shipping agent location that is closest to the sender's address (does not apply to shipping products with pickup). Is only sent if agents_in is true. Data type: STRING Max chars: 72 Required: No
<agent_in_link> **https://www.fraktjakt.se/agents/search_closest/377482?type=8&shipper=4** </agent_link>	A link to a map that shows where the shipping agent defined in <agent_in_info> is located. Is only sent if agents_in is true. Data type: STRING Max chars: 256 Required: No
- <shipper>	The shipper delivering the shipping product. This element (and its sub elements) is only created if <*shipper_info*> has been sent in the initial API call with the value *"1"*. Required: No
<id>4</id>	The shipper's ID # in Fraktjakt. Data type: INTEGER Required: No
<name>Bussgods</name>	The shipper's name. Data type: STRING Max chars: 35 Required: No
<logo_url> https://www.fraktjakt.se/images/shippers/4.png </logo_url>	Transportörens logo. I fall du vill visa The shipper image logo url. In case you want to present it in the search result. Datatype: STRING Max chars: 72 Required: No
</shipper>	
</shipping_product>	

`  </shipping_products>`	
` </shipment>`	

Order API

When it's time to create a shipping order, i.e. purchase shipping products that you found using the Query API, you need to send a new call to Fraktjakt. The calls that are used for ordering shipping products are collectively called the Fraktjakt Order API, or just ”Order API” in this document.

In many webshop platforms, such as OsCommerce, the shipping is purchased immediately after a webshop customer completes payment for items that the customer is purchasing in the webshop. However, other webshop platforms may purchase shipping products at a different point in the sales process.

Use the following URI to send an order to Fraktjakt via the Order API:

https://testapi.fraktjakt.se/orders/order_xml	(TEST API – test environment)
https://api.fraktjakt.se/orders/order_xml	(PROD API – live production environment)

Avoid using https://www.fraktjakt.se. That could seriously slow down your application and possibly lead to incorrect results.

IMPORTANT: If you create an order using a *shipment ID* that has already been used in a previous order, a completely new shipment and order will be created. The new shipment ID and Order ID are included in the XML that is returned by the *Order API*.

Call types

There are two different types of Order API calls:

Calltype 1 – Create an order from a previously created shipment

Usually the previously created shipment was created using the *Query API.*

The incoming XML contains the element, <shipment_id>, which identifies the previously created shipment.

Do not use the element, <parcels>, for this call type. The previously created original shipment (identified with <shipment_id> should already have sent in dimensions (length, width and height) for each package in it. The API will not allow an order to be created if the dimensions are missing in the shipment from which it's created.

If you send a call to Order API containing a shipment_id from a previous order, a new order will be created with a new shipment (with a new shipment_id). These will look the same as the previous shipment and order. This makes it easy to do multiple purchases of identical shipments with the same addresses.

Order API divides a shipment in to multiple shipments when the original shipment contains more than one package and the shipping service is one where the receiver collects the shipment at a service point.

It's possible to send addresses with the elements, <address_from> and <address_to>, but these will not be used here. The address will always be from the given shipment.

To create an order from a previous shipment-search the complete dimensions (length, width and height) must have been sent in the original shipment-search.

Call type 2 – Create and order directly without a previously created shipment.

This call type allows you to directly place an order in Fraktjakt.

The incoming XML does not contain a <shipment_id> element. Instead it contains the <parcels> element.

The alternatives for the <shipping_product_id> element can be found by calling https://www.fraktjakt.se/shipping_products/xml_list,

Address management:

- Sender address – If the sender address is sent with the element, <address_from>, it will be used, otherwise the registered delivery address will be used.
- Receiver address - If the receiver address is sent with the element, <address_to>, it will be used as the receiver address. Otherwise the sender address will be used. In most cases this is not desirable.

XML for Order API calls

XML element tag and example of contents	Description
<?xml version="1.0" encoding="UTF-8"?>	XML header
- <OrderSpecification>	This is the document element, i.e. the element that contains the entire order that is being sent to Fraktjakt. It contains one shipment. Required: Yes
<value>**10.2**</value>	[Currently not used] This is the value of all items in the order, i.e. the reimbursement amount in case of loss in transport. It should normally be the same as the shipment's value (See below.). Data type: FLOAT Required: No
- <consignor>	Information about the consignor, i.e. the webshop or other entity that is sending the order.
See consignor element on page 16.	
</consignor>	
<referred_consignor>	Use this tag if the shipment order will be finished and paid for by someone other than the consignor. Enter the name of the The response will then also include the tag <payment_link> that will link to the payment-page. Data type: INT Required: **No**

<id>**12121**</id>	The payer's customer ID in Fraktjakt. Data type: INT Required: **Yes, if referred_consignor is used**
<key>**121212121212121212121212121**</key>	The payer's customer ID in Fraktjakt. Data type: STRING Max chars: 64 Required: **Yes, if referred_consignor is used**
</referred_consignor>	
<shipment_id>**67887**</shipment_id>	Fraktjakt's shipment ID. This must be an ID that you have received in a response from the Query API. If you have already used the same shipment ID in a previous call to the Order API, Fraktjakt will create a new copy of the existing shipment record with a new shipment ID (but with the same shipment details). Data type: INT Required: Yes
<shipping_product_id>**15**</shipping_product_id> *(Should be one of the shipping product IDs from the response to a previous query_xml call.)*	The shipping product's ID in Fraktjakt. This must be an ID that you have received in a response from the Query API. Data type: INT Required: Yes
<anonymous_sender>**Another name**</anonymous_sender>	Used if you want to put another name as the sender on the shipping documents. Useful if the receiver don't wan't to show where their parcels comes from or you are arranging the shipment for someone else. Data type: STRING Required: No
<reference>**To my mother**</reference>	Free text that refers to the order in Fraktjakt. This element may contain your own system's order ID, an identifying text, or some other. This text will appear on your shipping labels. Data type: STRING Max chars: 50 Required: No
<export_reason>**SALE**</export_reason> *Valid values:* • *SALE – Any shipment being sold or sent free of charge, from one business entity to another, for the purpose of being re-sold or otherwise consumed in the course of carrying out its business.* • *GIFT – Any shipment containing articles to be given as an unsolicited gift, from one private individual to another private individual, which are not being shipped by a business nor consigned to a business.* • *SAMPLE – Any shipment containing articles which are sent free of charge that have been marked or mutilated or otherwise made unsuitable for sale or use except as commercial samples, being shipped with the view of soliciting an order or as a mock-*	*Reason for export* tells the purpose of the shipment. Data type: STRING Max chars: 16 Default: SALE Required: **International shipments**

up to furthering an existing order from the foreign entity. • *RETURN – The cargo that was bought from the receiver* • *REPAIR – Articles being shipped for repair or articles previously exported for repairs which are being returned following the repair.* • *PERSONAL EFFECTS – Any shipment containing used personal articles such as unaccompanied baggage and household goods being shipped for relocation.*	
<commodities>	The types of commodities that the shipment contains.
- <commodity>	A single commodity type. At least one commodity type must be included. **Minimum count = 1** **Excluded if query type 2 only if ordering a *Fraktbag*** (shipping_product_id = 137-141)
<name>**shoes**</name>	The name of the commodity type. Data type: STRING Max chars: 64 Required: Yes The value should be understandable in the receiving counrty
<quantity>**3**</quantity>	The quantity of commodities of this type. Measured in <quantity_units>-type Data type: INT Required: Yes
<taric>**0123456789**</taric> *(see http://tulltaxan.tullverket.se/#!/taric/nomenclature/sbt)*	The harmonized customs code for this commodity type (also called TARIC code). Data type: INT Required: No**, but when you include this element in international shipping orders, the shipments will normally get through customs much faster.**
<quantity_units>**EA**</quantity_units>	The unit of measure for this commodity type. Data type: STRING Max chars: 2 Required: No**, but it should be included in international shipping orders.** Default: EA (each) Allowed values: EA = each DZ = dozens L = liter ML = milliliter KG = kilogram
<description>**Blue suede leather shoes, size 43**</description> *(Attention! 15 characters minimum. The value should be understandable in the receiving country.)*	Description of the commodity type. Data type: STRING Min chars: 15 Max chars: 128 Required: **Yes when the shipment is international** The value should be understandable in the receiving country.

<country_of_manufacture>**US**</country_of_manufacture> *(Default is "SE" if no value is provided)*	Landskod, enligt ISO 3166-1 alpha-2 (5) standarden. Data type: STRING Max chars: 2 Default: SE Required: No**, men kan anges i frakt till utlandet** See Appendix 2 page 77 for the codes
<weight>**13.1**</weight> *(Attention! You may use a decimal point but not a comma.)*	The cummulative weight in kg of all items of this type in the shipment. The sum of all commoditiy weights must be equal to the sum of the weights of all items in the shipment. Data type: FLOAT Required: **Yes when the shipment is international**
<unit_price>**99.99**</unit_price> *(Attention! You may use a decimal point but not a comma.)*	The value per unit of this commodity type. It is used in customs and for complaint. Data type: FLOAT Required: **Yes when the shipment is international**
</commodity>	
</commodities>	
- <parcels>	Packages. At least 1 parcel. We do not recommend more than 10 parcels. MinOccur = 1 **for call type 2.** **Omitted with call type 2 if a *Fraktbag* is ordered** (shipping_product_id = 137-141). **Always omitted with call type 1**
- <parcel>	Package data
<weight>**3.2**</weight>	Package in weight Datatype: FLOAT Obligatory: Yes (if parcel is used)
<length>**35**</length>	Package length in cm Datatype: FLOAT Obligatory: Yes (if parcel is used) widest side
<width>**23.5**</width>	Paket bredd i cm Datatyp: FLOAT Obligatorisk: Yes (if parcel is used) second widest side
<height>**15**</height>	Package height in cm Datatype: FLOAT Obligatory: Yes (if parcel is used)
</parcel>	
</parcels>	
- <address_from>	Sender address Obligatory: No If not entered the address from the settings will be used.

See address element on page ?	
</address_from>	
- <address_to>	Receiver address Alternative name is 'address' Obligatory: yes, with call type 2
See address element on page ?	**Know that <language> can be used**
</address_to>	
- <recipient>	The receiver of the shipment. Required: Yes
<company_to>**Buy and Sell Ltd**</company_to>	The recipient's company name (if the recipient is a company). Data type: STRING Max chars: 32 Required: **Not if name_to is defined instead. Otherwise yes.**
<name_to>**Sven Andersson**</name_to>	The recipient's name (if the recipient is not a company). Data type: STRING Max chars: 32 **Required: Not if company_to is defined instead. Otherwise yes.**
<telephone_to>**0730710220**</telephone_to>	The recipient's telephone number. Data type: STRING Max chars: 32 Required: **No – but we recommend you to send it**
<mobile_to>**0730710220**</mobile_to>	The recipient's cell phone number. Data type: STRING Max chars: 32 Required: No
<email_to>**sven@webbutikenab.se**</email_to>	The recipient's e-mail address. Data type: STRING Max chars: 64 Required: Should be entered so that email notification works, including the tailor made notification from Fraktjakt.
</recipient>	
<sender_email>**info@example.info**</sender_email>	**[Only applies to pre-paid shipping]** The e-mail address of the person who will be handling the shipment, if not the consignor. This will generate a sender_email_link element in the reply and an email to that address when the order is paid. (7) Data type: STRING Max chars: 64 Required: No Fraktjakt automatically determines if the shipment is “receiver shipment” or “third party shipment” based on which addresses have been.
<booking>	Booking information. Shipments are always booked from the sender's

	address. If no booking address is included, Fraktjakt will use the address information that you entered on the webshops configuration page at Fraktjakt.se. Required: No
<driving_instruction>**Here and there**</driving_instruction>	Instructions on how to find the place where the shipment is to be picked up for delivery. Data type: String Required: No
<user_notes>**Port code 1112**</user_notes>	Notes to the deliverer with information that are needed in order for the booking to be carried out. Data type: String Required: No
<pickup_date>**2014-12-10**</pickup_date>	The date when the shipment is to be picked up. This must be a date in the future. Data type: Date (YYYY-MM-DD) Required: No
</booking>	
</OrderSpecification>	

Notes:

1. A shipment contains one or more parcels.
2. En shipment contains one or more commodity types.
3. The sum of the weights of all commodities in a shipment must be equal to the sum of the weights of all parcels in the shipment.
4. At least one contact method must be included for the shipping recipient. It is recommended that the recipient's e-mail address is always included.
5. https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2
6. Currently, only 'sv' is supported. For a description of the standard nomenclature for country names, see https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
7. Don't apply to services that are paid at the shippers own page/site, such as Schenker Privatpaket.

Once Fraktjakt receives an order via the Order API, the order is “completed” internally in Fraktjakt. This entails that the order is associated with your Fraktjakt account, pending payment. You will see the order in the tab labeled *”Orders”*.

Example – Call type 1 Domestic

Create an order from an earlier created shipment (usually created through *Query API*).

This XML will work if your replace *consignor id* and *key* to your own webshop values, as well as entering an existing *shipment_id*.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderSpecification>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>utf-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <referred_consignor>
    <id>REFERRED_CONSIGNOR_ID</id>
    <key>REFERRED_CONSIGNOR_KEY</key>
  </referred_consignor>
  <shipment_id>87566</shipment_id>
  <shipping_product_id>25</shipping_product_id>
  <reference>My brothers shoes</reference>
  <commodities>
    <commodity>
     <name>skor</name>
     <quantity>2</quantity>
    </commodity>
  </commodities>
  <recipient>
    <name_to>Olle Klint</name_to>
    <company_to>Hanson & Greve AB</company_to>
    <telephone_to>036190220</telephone_to>
  </recipient>
  <booking>
    <pickup_date>2014-12-10</pickup_date>
    <driving_instruction>Drive up on the alley to your right.</driving_instruction>
  <user_notes>Port code 1112</user_notes>
  </booking>
</OrderSpecification>
```

Example – Call type 1 International

Create an order from an earlier created INTERNATIONAL shipment (usually created through *Query API*).

This XML will work if your replace *consignor id* and *key* to your own webshop values, as well as entering an existing *shipment_id*.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderSpecification>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>utf-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <shipment_id>694556</shipment_id>
  <shipping_product_id>119</shipping_product_id>
  <reference>My brothers shoes</reference>
  <export_reason>GIFT</export_reason>
  <commodities>
    <commodity>
     <name>Skor</name>
     <quantity>2</quantity>
     <taric>0123456789</taric>
     <quantity_units>EA</quantity_units>
     <description>Black leather shoes with laces</description>
     <country_of_manufacture>FI</country_of_manufacture>
     <weight>2.8</weight>
     <unit_price>99.99</unit_price>
    </commodity>
  </commodities>
  <recipient>
    <name_to>Olle Klint</name_to>
    <company_to>Hanson & Greve AB</company_to>
    <telephone_to>036190220</telephone_to>
  </recipient>
  <booking>
    <pickup_date>2014-12-10</pickup_date>
    <driving_instruction>Drive up on the alley to your right.</driving_instruction>
    <user_notes>Port code 1112</user_notes>
  </booking>
</OrderSpecification>
```

Example – Call type 2 Domestic

Create an order directly *(without having created a shipment in an earlier query call)*.

This XML will work if your replace *consignor id* and *key* to your own webshop values, as well as entering an existing *shipment_id*.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderSpecification>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>utf-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <shipping_product_id>25</shipping_product_id>
  <reference>Ordernr 123456</reference>
  <commodities>
    <commodity>
     <name>Jeans</name>
     <quantity>1</quantity>
     <description>Blåjeans size 33x32</description>
    </commodity>
  </commodities>
  <parcels>
    <parcel>
      <weight>2.8</weight>
      <length>25</length>
      <width>20</width>
      <height>15</height>
    </parcel>
  </parcels>
  <address_from>
    <street_address_1>Gjuterigatan 9</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>55318</postal_code>
    <city_name>Jönköping</city_name>
    <residential>0</residential>
    <country_code>SE</country_code>
  </address_from>
  <address_to>
    <street_address_1>Hedenstorp 10</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>33292</postal_code>
    <city_name>Gislaved</city_name>
    <residential>1</residential>
    <country_code>SE</country_code>
  </address_to>
  <recipient>
    <name_to>Olle Klint</name_to>
    <telephone_to>036190220</telephone_to>
  </recipient>
  <booking>
    <pickup_date>2014-04-06</pickup_date>
    <driving_instruction>Drive up on the alley to your right.</driving_instruction>
    <user_notes>Port code 1112</user_notes>
  </booking>
</OrderSpecification>
```

Example – Call type 2 International

Create an order directly *(without having created a shipment in an earlier query call)*.

This XML will work if your replace *consignor id* and *key* to your own webshop values, as well as entering an existing *shipment_id*.

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderSpecification>
  <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>utf-8</encoding>
    <api_version>3.1.0</api_version>
  </consignor>
  <shipping_product_id>119</shipping_product_id>
  <reference>Ordernr 123456</reference>
  <export_reason>GIFT</export_reason>
  <commodities>
    <commodity>
     <name>Jeans</name>
     <quantity>2</quantity>
     <taric>0123456789</taric>
     <quantity_units>EA</quantity_units>
     <description>Levis blue jeans, size 33x32</description>
     <country_of_manufacture>FI</country_of_manufacture>
     <weight>1.5</weight>
     <unit_price>99.99</unit_price>
    </commodity>
  </commodities>
  <parcels>
    <parcel>
      <weight>1.5</weight>
      <length>25</length>
      <width>20</width>
      <height>15</height>
    </parcel>
  </parcels>
  <address_from>
    <street_address_1>Gjuterigatan 9</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>55318</postal_code>
    <city_name>Jönköping</city_name>
    <residential>0</residential>
    <country_code>SE</country_code>
  </address_from>
  <address_to>
    <street_address_1>101 Main Street</street_address_1>
    <street_address_2></street_address_2>
    <postal_code>12345</postal_code>
    <city_name>Schenectady</city_name>
    <residential>1</residential>
    <country_code>US</country_code>
    <country_subdivision_code>NY</country_subdivision_code>
  </address_to>
  <recipient>
    <name_to>Jack Martins</name_to>
    <telephone_to>(732)555-1212</telephone_to>
  </recipient>
  <booking>
    <pickup_date>2014-04-06</pickup_date>
    <driving_instruction>Drive up on the alley to your right.</driving_instruction>
    <user_notes>Port code 1112</user_notes>
  </booking>
</OrderSpecification>
```

Response

XML element tag and example of contents	Description
<?xml version="1.0" encoding="UTF-8"?>	
- <result>	
<status>**ok**</status>	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
<code>**1**</code>	0=OK 1=warning 2=Error
<warning_message>**You need to define all measurements**</warning_message>	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
<error_message></error_message>	Error message. Data type: STRING Max chars: 500 Required: Yes
<shipment_id>**45654**</shipment_id>	Fraktjakts shipment ID (1) Data type: INT Required: Yes
<access_code>**ABC12345**</access_code>	A code for accessing and managing the shipment without previously being logged in. (3)
<access_link> https://www.fraktjakt.se/shipments/show/163221?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9 </access_link>	A link to the shipment using the above SHIPMENT_ID and ACCESS_CODE. Data type: STRING Required: No
<order_id>**67887**</order_id>	Fraktjakt's order ID Data type: INT Required: Yes
<amount>**1066.33**</amount>	The cost of the shipment, excluding VAT. Data type: FLOAT Required: Yes
<currency>**SEK**</currency>	Currency corresponding to the shipment cost. Data type: STRING Max chars: 3 Required: Yes
<payment_link> https://**www.fraktjakt.se/orders/finish/1631?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9** </payment_link>	This tag contains a link to the payment page If a referred_consignor tag is submitted for the paying user. (2) Data type: STRING Required: No

<sender_email_link> https://www.fraktjakt.se/orders/confirmation/1671?key=cabec52762637412652e5023deb5e5c2 </sender_email_link>	A link to handle the shipment after the order is paid. The tagg is added if sender_email is sent in the order-call. Data type: STRING Required: No
</result>	

Notes:

1. This is not necessarily the same *shipment_id* that was returned when creating the order, since that *SHIPMENT_ID* can be reused and changed if the shipment is split into several new shipments.

   This *SHIPMENT_ID* can be used to track the shipment with one of these links:
   (just substitute *SHIPMENT_ID* with the actual shipment_id number that was returned)

   https://www.fraktjakt.se/trace/list_shipment/SHIPMENT_ID

   https://www.fraktjakt.se/trace/xml_trace?consigner_id=YOUR_CONSIGNOR_ID&consigner_key=YOUR_CONSIGNOR_KEY&shipment_id=SHIPMENT_ID

2. See also *Track & Trace API* for more information about this

3. Available if <api_version> is larger than 2.9.2. Put the code in link:
   https://www.fraktjakt.se/shipments/show/SHIPMENT_ID&access_code=ACCESS_CODE
   For more information about the link, see page 59

Shipment API (Merchant controlled)

This API creates a shipment in Fraktjakt. No shipment price search is done, therefore the webshop administrator must do the shipment search later in Fraktjakt.

With this API, a call is done from the webshop to Fraktjakt when a purchase is made in the webshop with the product data (dimensions and weight) and addresses so that you can search and purchase a shipment later in Fraktjakt.

The shipment must have Sweden as the receiver or sending country, or both.

Prepare a shipment in Frakjakt by calling Shipment API with the following URL:

https://testapi.fraktjakt.se/shipments/shipment_xml (TEST API – test environment)

https://api.fraktjakt.se/shipments/shipment_xml (PROD API – production environment)

The shipment can be searched using the following URL:

https://www.fraktjakt.se/shipments/show/SHIPMENT_ID&access_code=ACCESS_CODE

The setting for email from Fraktjakt called 'Incoming shipment connection' in the webshop's settings in Fraktjakt determines if a mail from Fraktjakt to the webshop admin should be sent for every call to this API.

Call

XML element tag and example of contents	**Description**
`<?xml version="1.0" encoding="UTF-8"?>`	XML-header
`- <CreateShipment>`	The call to Fraktjakt Obligatory: **Yes**
`  - <consignor>`	Information about who's making the call.
See Consignor-element on page 17	
`    </consignor>`	
`  - <referred_consignor>`	Use this element if someone other than the consignor will be paying for the transportation. The element should contain the name of the person who will be handling the shipment. The response will then also include the element, <payment_link>, that will link to the payment-page. Data type: INT
`      <id>REFERRED_`**`CONSIGNOR_ID`**`</id>`	The payer's customer ID in Fraktjakt. Data type: INT Required**: Yes, if referred_consignor is used**

`<key>`**`REFERRED_CONSIGNOR_KEY`**`</key>`	The payer's customer key in Fraktjakt. Data type: STRING Max chars: 64 Required: **Yes, if referred_consignor is used**
`</referred_consignor>`	
`<reference>`**`Orderid 1236`**`</reference>`	Individual field for referencing the shipment in Fraktjakt. It could be your own order_id, help text or any value you choose. Also visible on all shipping notes. Datatype: STRING Max chars: 50 Only digits, letters A-Z and space are allowed.. Obligatory: No
`- <address_from>`	Sender address Obligatory: No If not entered the address from the settings will be used.
`See address element on page ?`	
`</address_from>`	
`- <address_to>`	Receiver address Obligatory: No If not entered, the sender address will be used. Which is usually not something you want.
`See address element on page ?`	**Know that <language> can be used**
`</address_to>`	
`<export_reason>`**`SALE`**`</export_reason>` *Valid values:* • *SALE – Any shipment being sold or sent free of charge, from one business entity to another, for the purpose of being re-sold or otherwise consumed in the course of carrying out its business.* • *GIFT – Any shipment containing articles to be given as an unsolicited gift, from one private individual to another private individual, which are not being shipped by a business nor consigned to a business.* • *SAMPLE – Any shipment containing articles which are sent free of charge that have been marked or mutilated or otherwise made unsuitable for sale or use except as commercial samples, being shipped with the view of soliciting an order or as a mock-up to furthering an existing order from the foreign entity.* • *RETURN – The cargo that was bought from the receiver* • *REPAIR – Articles being shipped for repair or articles previously exported for repairs which are being returned following the repair.* • *PERSONAL EFFECTS – Any shipment containing used personal articles such as unaccompanied baggage and household goods being shipped for relocation.*	*Reason for export* tells the purpose of the shipment. Data type: STRING Max chars: 16 Default: SALE Required: **International shipments**
`- <recipient>`	The receiver of the shipment. Required: Yes
`<company_to>`**`Köp och sälj AB`**`</company_to>`	The recipient's company name (if

	the recipient is a company). Data type: STRING Max chars: 32 Required: **Not if name_to is defined instead. Otherwise yes.**
`<name_to>`**`Sven Andersson`**`</name_to>`	The recipient's name (if the recipient is not a company). Data type: STRING Max chars: 32 **Required: Not if company_to is defined instead. Otherwise yes.**
`<telephone_to>`**`036190220`**`</telephone_to>`	The recipient's telephone number. Data type: STRING Max chars: 32 Required: **No – but we recommend you to send it**
`<mobile_to>`**`0709999999`**`</mobile_to>`	The recipient's cell phone number. Data type: STRING Max chars: 32 Required: No
`<email_to>`**`sven@webbutikenab.se`**`</email_to>`	The recipient's e-mail address. Data type: STRING Max chars: 64 Required: Should be entered so that email notification works, including the tailor made notification from Fraktjakt.
`</recipient>`	
`<commodities>`	The types of commodities that the shipment contains. MinOccur = 1
`- <commodity>`	Commodity data
`<name>skor</name>`	The name of the commodity type. Data type: STRING Max chars: 64 Required: Yes
`<quantity>`**`3`**`</quantity>`	The quantity of commodities of this type. Measured in <quantity_units>-type Data type: INT Required: Yes
`<taric>0123456789</taric>` *(se http://tulltaxan.tullverket.se/#!/taric/nomenclature/)*	The harmonized customs code for this commodity type (also called TARIC code). Data type: INT Required: No**, but when you include this element in international shipping orders, the shipments will normally get through customs much faster.**
`<quantity_units>`**`EA`**`</quantity_units>`	The unit of measure for this commodity type. Data type: STRING Max chars: 2 Required: No**, but it should be included in international shipping orders.** Default: EA (each) Allowed values: EA = each

	DZ = dozens
	L = liter
	ML = milliliter
	KG = kilogram

<description>**Blue suede leather shoes, size 43**</description> *(Attention! 15 characters minimum. The value should be understandable in the receiving country.)*	Description of the commodity type. Data type: STRING Min chars: 15 Max chars: 128 Required: **Yes when the shipment is international** The value should be understandable in the receiving country.
<country_of_manufacture>**US**</country_of_manufacture> *(Default is "SE" if no value is provided)*	Landskod, enligt ISO 3166-1 alpha-2 (5) standarden. Data type: STRING Max chars: 2 Default: SE Required: No**, men kan anges i frakt till utlandet** See Appendix 2 page 77 for the codes
<weight>13.1</weight>	The cummulative weight in kg of all items of this type in the shipment. The sum of all commoditiy weights must be equal to the sum of the weights of all items in the shipment. Data type: FLOAT Required: **Yes when the shipment is international**
<unit_price>**99.99**</unit_price>	The value per unit of this commodity type. Is used in customs and for complaint. Data type: FLOAT Required: **Yes when the shipment is international**
</commodity>	
</commodities>	
</CreateShipment>	

Example – Domestic

```
<?xml version="1.0" encoding="UTF-8"?>
  <CreateShipment>
    <consignor>
    <id>YOUR_CONSIGNOR_ID</id>
    <key>YOUR_CONSIGNOR_KEY</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
    <system_name>Workpress</system_name>
    <module_version>1.0</module_version>
    <api_version>3.1.0</api_version>
  </consignor>
    <reference>Test shipment</reference>
    <address_to>
      <street_address_1>Brunnsgatan 16</street_address_1>
      <street_address_2></street_address_2>
      <postal_code>55317</postal_code>
      <residential>0</residential>
      <country_code>SE</country_code>
      <language>sv</language>
    </address_to>
    <recipient>
      <company_to>Fraktjakt AB</company_to>
      <name_to>Customer Service</name_to>
      <telephone_to>0709-399981</telephone_to>
      <email_to>testmail@fraktjakt.se</email_to>
    </recipient>
    <commodities>
      <commodity>
        <name>Cell phone</name>
        <quantity>3</quantity>
        <taric>01234567</taric>
        <quantity_units>EA</quantity_units>
        <description>Apple iPhone X</description>
        <country_of_manufacture>US</country_of_manufacture>
        <weight>0.6</weight>
        <unit_price>2395</unit_price>
      </commodity>
    </commodities>
  </CreateShipment>
```

Svar

XML element tag and example of contents	**Description**
`<?xml version="1.0" encoding="UTF-8"?>`	
- <result>	
<status>**ok**</status>	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
<code>**1**</code>	0=OK 1=warning 2=error
<warning_message>**Du behöver ange alla mått**</warning_message>	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
<error_message></error_message>	Error message. Data type: STRING Max chars: 500 Required: No
<shipment_id>**45654**</shipment_id>	Fraktjakts shipment ID Data type: INT

	Required: Yes
`    <access_code>ABC12345</access_code>`	A code for reaching and managing the shipment without being logged in.

`    <access_link>`
`https://www.fraktjakt.se/shipments/show/163221?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9`
`    </access_link>`
	A link to the shipment using the above SHIPMENT_ID and ACCESS_CODE.
	Data type: STRING
	Required: No

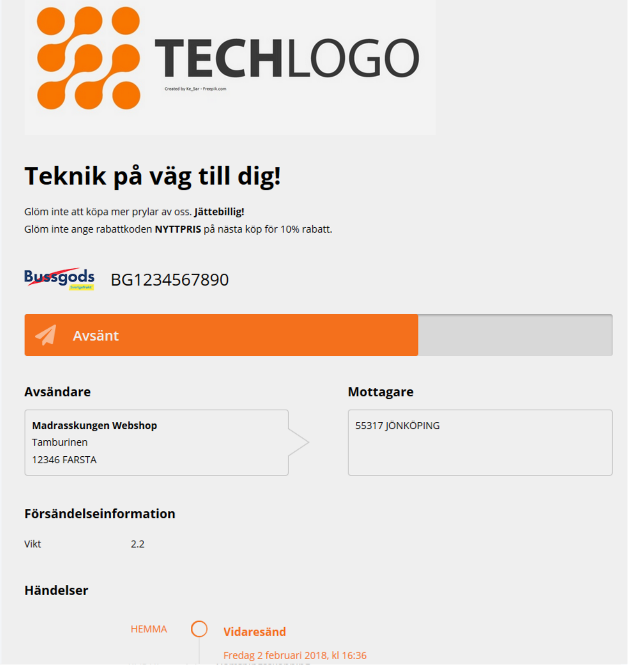
`  </result>`	

After the call you need to access the new shipment using this link:

https://www.fraktjakt.se/shipments/show/SHIPMENT_ID&access_code=ACCESS_CODE

Track & Trace API

In Fraktjakts settings there's a tool for creating your own page for tracking shipments. This is an example of how it could look.



Teknik på väg till dig!

Glöm inte att köpa mer prylar av oss. **Jättebillig!**
Glöm inte ange rabattkoden **NYTTPRIS** på nästa köp för 10% rabatt.

BG1234567890

Avsänt

Avsändare

Madrasskungen Webshop
Tamburinen
12346 FARSTA

Mottagare

55317 JÖNKÖPING

Försändelseinformation

Vikt 2.2

Händelser

HEMMA ○ **Vidaresänd**
Fredag 2 februari 2018, kl 16:36

If the settings provided doesn't suit your demands you could create a completely new site or get the tracking information directly to your site with this API.

This API is used to quickly be able to present the status of a shipment for your users. It makes a very quick query to Fraktjakt and returns the response as a XML. The information has Fraktjakt collected from the shippers. It might therefor be a bit old.

The Call

Please note that no XML is needed for the call. We wanted to make the call as easy as possibly.

Track a shipment in Frakjakt by calling Track & Trace API with the following URL:

https://testapi.fraktjakt.se/trace/xml_trace?consigner_id=YOUR_CONSIGNOR_ID&consigner_key=YOUR_CONSIGNOR_KEY&shipment_id=SHIPMENT_ID (TEST API – test environment)

https://www.fraktjakt.se/trace/xml_trace?consigner_id=YOUR_CONSIGNOR_ID&consigner_key=YOUR_CONSIGNOR_KEY&shipment_id=SHIPMENT_ID (PROD API – production environment)

SHIPMENT_ID is the shipment_id returned from the *Order API*. It is not the same *shipment_id* that is returned from the *Query API*. Those two id's might be different.

Also be aware that Fraktjakt might split one shipment into several shipments and that each one of these might have a different status. This is done for all shipping products that require the customer to pick up the package at a service point.

Example

This HTTP call should work if you switch YOUR_CONSIGNOR_ID and YOUR_CONSIGNOR_KEY to your own and supply the SHIPMENT_ID from the response to one of your previous API calls.

http://testapi.fraktjakt.se/trace/xml_trace?consigner_id=YOUR_CONSIGNOR_ID&consigner_key=YOUR_CONSIGNOR_KEY&shipment_id=SHIPMENT_ID

Response

XML element tag and example of contents	Description
`<?xml version="1.0" encoding="iso-8859-1"?>`	
`- <result>`	
`<status>`**ok**`</status>`	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
`<code>`**0**`</code>`	0=OK 1=warning 2=Error
`<warning_message></warning_message>`	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
`<error_message></error_message>`	Error message. Data type: STRING Max chars: 500 Required: Yes
`<shipping_states>`	The status of the different shipments MinOccur = 1
`<shipping_state>`	Status for one of the shipments MinOccur = 0 (If the wrong shipment_id is sent)
`<shipment_id>`**204025**`</shipment_id>`	Id for the shipment. Might be a different one than the one sent in

	the call. Datatyp: INTEGER Required: Yes
`<name>`**Not payed**`</name>`	Status as text. Datatyp: STRING Max chars: 500 Required: Yes
`<id>`**0**`</id>`	Status as a number (1). Datatyp: INTEGER Required: Yes
`<fraktjakt_id>`**8**`</fraktjakt_id>`	Fraktjakt's internal status as a number Datatyp: INTEGER Required: Yes
`</shipping_state>`	
`</shipping_states>`	
`</result>`	

Status Codes

The status of a shipment is seen in the *fraktjakt_id* field that corresponds to the shipping_state of the shipment identified by *shipment_id.*

The following table shows the meanings of the numbers in <shipping_state> elements.

<fraktjakt_id>	<id>	<name>	Comments
0	0	Not paid	If a shipment_id from a Query API response was sent and an Order API call did not send it.
1	0	Preliminary	A shipment_id was sent with an Order API call and a preliminary order has been created.
2	0		Not in use
3	0	Paid	The order for the shipment has been paid for. This means that one or more of the following things happened, depending on the shipping service: • Shipping documents have been created • EDI have been sent to the carrier • Pickup at the sender is booked • Email sent to the sender.
12	0	Handled by the carrier	The shipment order is present in Fraktjakt but the payment will be made externally at the carrier. An example of this is when you buy Schenker Privpaks services.
4	1	Sent	The carrier has received the shipment either by pickup at the sender or if the sender left it at the terminal or the delivery agent.
5	2	Delivered	Sändningen har levererats till mottagaren eller till ett utlämningsombud nära till mottagaren, beroende på frakttjänsten som har köpts. The Shipment was delivered to the receiver or to a delivery point close to the receiver, depending on the shipping service purchased.
6	3	Signed	The receiver has signed the delivery of the shipment.
7	4	Returned	The shipment has been returned to the sender for some reason.
18	0	On hold	The shipment was created through order import and is awaiting processing.

Link to the shipment

When a shipment has been created through an API it cat always be accessed on Fraktjakt.se if you have an access_code connected to that shipment. It's retrieved from the Order-API and Shipment-API response.

The shipment can be accessed with this link:
https://www.fraktjakt.se/shipments/show/SHIPMENT_ID&access_code=ACCESS_CODE

If the shipment search isn't complete you're directed to a page where you can complete it. If it's not purchased you're directed to the cart and if it is purchased you get to a page with the shipment or an order confirmation if you're not logged in.

Webshops API (create a user and a webshop)

It can be practical to automatic create a user account and webshop information in Fraktjakt if you want to integrate a webshop platform with Fraktjakt. You can send in your users account-information to Fraktjakt and create accounts in both systems. It is relatively simple to do that. All you have to supply the API with is login, email address and telephone number to the new user.

To create a user-account and a webshop in Fraktjakt, use the API that has the following URL:

https://testapi.fraktjakt.se/webshops/create_xml	(TEST API – test environment)
https://api.fraktjakt.se/webshops/create_xml	(PROD API – production environment)

IMPORTANT The values you don't send in for the user and the webshop are copied from your user, except the password that will be set to a random string.
No accounts can be used before they are activated. You don't have to create a webshop. This API can also be used to just create a user account. All settings for access rights, webshop prices and what services that are shown are copied from your user.

XML for webshop API calls

XML element tag and example of contents	Description
<?xml version="1.0" encoding="iso-8859-1"?>	
- <webshop>	
- <consignor>	Information about the consignor, i.e. the webshop or other entity that is sending the order.
See consignor-element on page 16	
</consignor>	
<email>**name@email.se**</email>	Users email Data type: STRING Required: Yes
<name>**Olle Klint**</name>	Users name (not login) Data type: STRING Required: No Default: Consignor-users value
<password>**secret**</password>	Users password Data type: STRING Required: No Default: Random
<login>**knut**</login>	Users login Data type: STRING Required: Yes
<telephone>**036123456**</telephone>	Users telephone number Data type: STRING Required: Yes
<new_webshop>	Block for the new webshop. No new webshop-information will be created if the block isn't provided. All values except the name are optional. If nothing is provided in a

	tag, the value are copied from the consignor-user.
<name>**Joppes Webshop**</name>	Webshop name Data type: STRING Required: Yes
<url>**https://kottegott.se**</url>	Webshop URL Data type: STRING Required: No Default: Consignor-users value
<ready_time>**07:15**</ready_time>	The earliest time that pickup is possible. This time should be as early as possible, and it must be in the future. Data type: Time (HH:MM) Required: **No**
<close_time>**16:16**</close_time>	The latest time that pickup is possible. This time should be as late in the afternoon as possible, and it must be in the future. Data type: Time (HH:MM) Required: No
<driving_instruction>**First drive here, then there, then you have arrived**</driving_instruction>	Direction to the site for pickup. Data type: STRING Required: No Default: Consignor-users value
<registration_number>**650616-7235**</registration_number>	Webshops registrationnumber. Needed for invoice Data type: STRING Required: No Default: Consignor-users value
</new_webshop>	
<sender_address>	Sender address for the webshop and the user.f. Used in alla fright-querys. Required: No Default:Consignor-users value
See Address-element on page ?	
</sender_address>	
<invoice_address> Address-element on page ? </invoice_address>	Invoice-address. The sender_address will be used If it isn't provided, or if that isn't provided, the invoice-address for the consignor-user. Required: No Default: Consignor-users value
<postal_address> See Address-element on page ? </postal_address>	Postal-address. The sender_address will be used If it isn't provided, or if that isn't provided, the postal-address for the consignor-user. Required: No Default: Consignor-users value
</webshop>	

response

XML element tag and example of contents	Description
`<?xml version="1.0" encoding="iso-8859-1"?>`	
`- <result>`	
`<status>`**ok**`</status>`	Status of API call ok = API call succeeded. warning = API call succeeded with warnings. error = API call failed. *Sent in response only for API versions 3.1.0 or greater.*
`<code>`**1**`</code>`	0=OK 1=warning 2=Error
`<warning_message>`**Du behöver ange alla mått**`</warning_message>`	Warning message with information that may be useful, especially when testing your integration (only in Swedish in the current version). Data type: STRING Max chars: 500 Required: Yes
`<error_message></error_message>`	Error message. Data type: STRING Max chars: 500 Required: Yes
`<user>`**Olle Klint**`</user>`	users name Datatyp: STRING Obligatorisk: **Yes**
`<id>`**12425**`</id>`	Users ID – Used in future call to Fraktjakt Data type: STRING Obligatorisk: **Ja**
`<key>`**83f0a2481e84506df46dd16fd6a94b538c4d511e**`</key>`	Users key – Used in future call to Fraktjakt Data type: STRING Obligatorisk: **Ja**
`</result>`	

Testing using the API Test Page

You may use the forms found at https://www.fraktjakt.se/xml/query_test to test your XML. You can find this page in the left-menu in the webshop settings called 'API Test page'.

Here you can try the following test forms for your XML-calls:

- Query API
- Requery API
- Order API, type 1
- Order API, type 2
- Shipment API
- Track & Trace API

There are several additional xml examples prepared with your Consignor ID & Key already in them (If you're logged in and you've selected your registered webshop), just click submit to try them out.

The above forms are also available in the PROD API live production environment, in case you have reason to test your XML documents there. **Be aware that any Order API documents that you enter in the PROD API environment will generate real shipping orders.**

To test another XML document, copy and paste the entire contents of the XML document directly in the form at one of the above links.
Then press the *"Submit"* button. This will cause the same results to be displayed in your browser that you would have received from the *Fraktjakt API* if you had sent the XML document via the API.

Error reporting

If you encounter problems when using the Fraktjakt API, you may send the entire XML document along with the response you received from the API to the following e-mail address: error_reports@fraktjakt.se.

We will promptly analyse the problem and respond to the reply-to address of the e-mail.

Feel free to automatically generate mails to error_reports@fraktjakt.se when errors occur, but please ensure that there is someone available to receive our responses to such error reports.

Common problems

- **Följande indata saknas - :xml**

  You have not submitted through the obligatory http-parameter – *'xml'* in your URL. The parameter consists of xml that contains all data necessary for the call.

- **My XML-file doesn't work no matter what I do!**

  Remember that it has to be URL-encoded before you send it.

  https://en.wikipedia.org/wiki/Query_string#URL_encoding

- **#<REXML::ParseException: No close tag for /shipment>**

  You have left out a closing tag for the element specified in the error text.

- **Consignor Id ej giltig . / Consignor Id ej angivet.**

  You have either forgot to include your Consignor ID or the current one isn't correct https://testapi.fraktjakt.se/webshops/install

- **Consignor nyckel (key) ej giltig. / Consignor nyckel (key) ej angivet.**

  You have either forgot to include your Consignor key or the current key isn't correct https://testapi.fraktjakt.se/webshops/install

- **Okänd frakttjänst-id angivna i taggen shipping_product_id.**

  The ID entered in <*shipping_product_id*> is not available in Fraktjakt. Enter a correct ID, like the one you got from your Query API call.

- **Frakttjänsten angivna i taggen shipping_product_id är icke köpbar.**

  The current shiping product for <shipping_product_id> is not currently purchasable in Fraktjakt. Contact Fraktjakt's support to get updated information about the matter.

- **Det går inte att avgöra vilken Order API anropstyp som används.**

  You have to include <*shipment_id*> and exclude <*parcels*> in calls for Order API type 1. In Type 2 however, you must include <*parcels*> and exclude <*shipment_id*>.

- **Webshoppen inte riktigt registrerad.**

  Your account either lacks a registered webshop or some required information about the webshop are missing.

- **Recipient måste anges.**

  An order cannot be created without the receiver name or company name in the elements <name_to> and <company_to>. One or the other have to be entered or both. There are other optional elements that can be sent in.

- **Vikten måste anges för paket 1. Textvärdet saknas.**

  The weight of the parcels has to be included in calls for *Query API* and *Order API* type 2.

- **Ordern kunde inte skapas: (Längden måste anges för paket 1)**
- **Ordern kunde inte skapas: (Bredden måste anges för paket 1)**
- **Ordern kunde inte skapas: (Höjden måste anges för paket 1)**

  The parcel length, width, height must be included in calls for *Order API* type 2.

- **Varuslagen/Innehållet måste anges.**

  The commodities must be entered in calls for the *Order API*. This is done with the element, <*commodities*>. The *name* and *amount* of at least one commodity has to be included in the <commodity> element.

- ~~**Fraktjakt hanterar inte i dagsläget några frakter skickat från utlandet.**~~

  ~~The sender address must be a swedish address since Fraktjakt currently doesn't support foreign senders. The sender address is entered into <address_from> in calls to Query API and Order API type 2.~~

- **No response from Fraktjakt**

  This problem is typically caused by a communication error. The most common sources of these errors are firewalls at web hotels. Verify that the firewall is permitting all traffic between your webshop and Fraktjakt.

  The problem may also be caused by a setting in your server that prohibits *cULR (libcurl)* from working properly. Try changing the communication method in your administration interface to *”fopen”*.

  Another possible cause of this problem is that Fraktjakt is experiencing a temporary communication problem. In order to eliminate this as the cause of the problem, verify that you can connect to https://testapi.fraktjakt.se (TEST API) or https://api.fraktjakt.se (PROD API) in a web browser.

  If you are experiencing problems ordering international shipping, the culprit could be an incorrect recipient address. Verify that your system has sent a valid recipient address to Fraktjakt.

  The sender's or recipient's address (or both) must be located in Sweden.

- **Incorrect server**

  Fraktjakt exposes two separate server addresses - one for the test environment (https://testapi.fraktjakt.se) and one for the production environment (https://api.fraktjakt.se). Make sure you are using a consignor ID and consignor key that are valid in the environment that you are communicating with.

- **You receive an XML file that contains the text, *”503 Service Temporarily Unavailable”*.**

There is probably a problem in the Fraktjakt server, and we are fixing the problem.

This problem may occur in the test environment, but it shouldn't be encountered in the production environment. The production environment has a very high degree of redundancy and failsafe measures.

- **The account that you were using in the test environment has stopped working**

  Once in a while, the test environment must be purged to clean out erroneous data that has been generated in countless tests. You will have to create a new test account when this happens.

- **Orders from my OsCommerce webshop are not reaching Fraktjakt.**

  If you have installed the OsCommerce module for Fraktjakt, you must make sure that both files in the installation package are installed. The first file only makes it possible to present shipping prices in your webshop. The second module (The Fraktdata module) allows orders to be sent to Fraktjakt.

- **Not all shipping products are shown in the results that are supposed to be shown.**

  Check your settings for your webshop if you are searching as a private person or as a company*(swedish: "företag")*.

- **I can't see any prices for my orders in Fraktjakts order-list.**

  You will not get any prices in the order-list if you test to search and buy shipments with the sama postal-code as you have entered for your webshops shipping-address (godsadress). Your prices will be recalculated and displayed when you tries to buy your order.

  Having the same sender- and receiver-postal-code is the way to tell Fraktjakt that you want to do a simplified search that is much faster, but don't save that much data.

  You can just take it easy. When you buy your orders for real, your will be able to pay for it :)

Tips when planning a parcel module integration

Parcel information

In order to find a shipper, it is important to provide information about the weight, dimensions and destination of the parcel(s) you will be sending. A list of the products/items you will be sending is not necessary. Minimum and standard dimensions can also be used.

Calculating minimum dimensions for a parcel with multiple items

Assume that every product has a given length, width and height and dimensions are therefore sorted in that order.

The longest side of the product will be considered as the length, the next longest will be the width and the shortest will be the height.

Create a parcel with a dimension of 0 (length), 0 (width) and 0 (height) = 0,0,0

Take the products (one at a time) that are to be placed in a parcel and do iteration (process repetition). Note: the parcel is the box you would use in sending products (items).

```
DO
Increase the parcel's shortest dimension with a product's shortest dimension. The parcel's shortest dimension is now longer.
Find the width of both the parcel and product. Set the parcel's width by using the highest value of the both.
Find the length of both the parcel and product. Set the parcel's length by using the highest value of the both.
Sort the parcel's dimensions through size (largest to smallest measurements where the length has the largest value, width has the second highest value and height has the shortest value). The parcel will most likely have a new shortest dimension.
END
```

Referring to exercise above, the parcel should now have a minimized dimension given the products that were to be placed in that parcel.

Note that re-sorting the dimensions is important with every new product addition to the parcel.

If a product utilizes bubble wrap or some other type of protective covering, it is recommended to use the size of the product with the protective covering.

Example:
Say that you have three products with the measurements 30x20x15, 40x15x15 and 30x30x23.
This will be the result.

Create a package with the measurements 0x0x0 cm

(product 30x20x15) Add the first product. The package will get the measurements 30x20x15. Sort them in order.
(product 40x15x15) Add the next products shortest side to the shortest side of the package and since the products longest side is longer then the package, the package gets that length. The package then becomes 40x20x30. Resort it and it becomes 40x30x20 (new shortest measuremen).
(product 30x30x23) Add the next products shortest side to the shortest side of the package. None of the other sides are longer then any side for the package. The package then becomes 40x30x43. Resort it and the package becomes 43x40x30.

The logic listed bellow can be used when calculating how the products can be divided to fit in a few packages. It's possible to program a more precise algorithm, bur this is outside of the scope of this document.

Notice! The following calculation procedure works best if the products length and width are at least half the size of the webshops standard packages. If your webshop don't have a lot of products that are ordered in large amounts it's better t o calculate how many fits laying down in the webshops standard package and treat them like a larger product with that size.

1. Create a table with the dimensions of the webshops standard cartons, in order from the largest to the smallest. Each shipment you create will contain 0 or more packages of the largest size, (the first standard package in the table), plus the last package in the shipment. One of the smaller standard cartons could be used for the last package, depending on the size of the contents.

Carton name	**Length**	**Width**	**Height**	**Empty Weight**
Large				
Medium				
Small				

2. Create a table with the products in the order, for calculating their package placement.

Product	**Product length**	**Product width**	**Product height**	**Product weight**	**Package weight**	**Package length**	**Package width**	**Package height**

3. Put in the product dimensions in the table, in descending order. The longest side of each product should be in the column Product length, the second longest in the Product width and the shortest side in Product height. The product should be in descending order from the one with the greatest length to the one with the shortest length. Products with the same length should be sorted by width and if they have the same width, sort them by height.

4. Type in the products weights in the column Product weight.

5. Copy the values in the columns *Product length, Product width, Product height and Product weight* in the first table row to the columns *Package length, Package width, Package height and Package weight* on the same row.

6. If there aren't any more products in the table skip ahead to step 9.

7. Type in the following three values in the columns *Package length, Package width and Package height on row 2,* sort them so that *Package length* contains the largest of the three values and *Package height* the smallest:
   - the largest of the *Package length* values on row 1 and *Product length* on row 2,
   - the largest of the *Package width* values on row 1 and *Product width* on row 2,
   - the sum of *Package height* on row 1 and *Product height* on row 2.

8. Write the sum of the *Package weight* on row 1 in the table and the *Product weight* on row 2 in *Package weight* on row 2.

9. Repeat step 6-8 for the remaining products in the table, one row at a time (Increase the row numbers mentioned in step 7 and 8 with 1 each time.) til all the products have been dealt with or package dimensions (*Package length, Package width* and *Package height*) of the this row gets larger than what fits in the webshops largest standard carton. If the latter happens the product in the previous row becomes the last in a package. So it's a package with weight in the column Package weight that's in the previous row and the dimensions (*Length, width, height)* that's in the first row of the table over the webshops standard cartons. Then replace the content in *Package weight, Package length, Package width and Package height* on this row with *Product weight, Product length, Product width and Product height* on the same row. Then use this row in step 6 instead of using row 1 and the next table row instead of row 2.

10. When you've done step 6-9 on the last product in the table the values in *Package weight, Package length, Package width and Package height* will be the smallest dimensions on the last package in the order. Change length, width and height to the dimensions of the smallest

standard package that's larger than these smallest dimensions.

11. Add the weight of the standard cartons to the weight of each package included in the shipment after the above calculation procedure is done.

When you've done the above the packages have their dimensions and weights, which you should put in these package definitions in the XML which you send to Fraktjakts Query API to get shipping alternatives for them.

Your implementation of this calculation procedure should also save all information needed to create a pick list.

It's also recommended to include the size of packing material and protective packing in the dimensions of the webshops products.

Code in PHP (thanks to Timmy Almroth - https://www.tim-international.net/)

```
<?php

// A fancy parcel function
  function calculate_parcel($products) {

  // Create an empty package
    $package_dimensions = array(0, 0, 0);

  // Step through each product
    foreach($products as $product) {

    // Create an array of product dimensions
      $product_dimensions = array($product['width'], $product['height'],
$product['length']);

    // Twist and turn the item, longest side first ([0]=length, [1]=width, [2]=height)
      rsort($product_dimensions, SORT_NUMERIC); // Sort $product_dimensions by highest to
lowest

    // Package height + item height
      $package_dimensions[2] += $product_dimensions[2];

    // If this is the widest item so far, set item width as package width
      if($product_dimensions[1] > $package_dimensions[1]) $package_dimensions[1] =
$product_dimensions[1];

    // If this is the longest item so far, set item length as package length
      if($product_dimensions[0] > $package_dimensions[0]) $package_dimensions[0] =
$product_dimensions[0];

    // Twist and turn the package, longest side first ([0]=length, [1]=width, [2]=height)
      rsort($package_dimensions, SORT_NUMERIC);
    }

    return $package_dimensions;
  }

// Set items
  $item1 = array(
    'width' => 30,
    'height' => 20,
    'length' => 15,
    'foo' => 'bar'
  );
  $item2 = array(
    'width' => 40,
    'height' => 15,
    'length' => 15,
    'foo' => 'bar'
  );
  $item3 = array(
    'width' => 23,
    'height' => 30,
    'length' => 30,
    'foo' => 'bar'
  );

// Run items through function
  $parcel = calculate_parcel(array($item1, $item2, $item3));

  var_dump($parcel);

?>
```

Create a fall-back in case of problems

Although we strive to maintain a stable environment for our system, there still are circumstances beyond our control which can affect the stability and access to the system. Examples could be power outage, loss of broadband connection, hardware failure, etc.

We therefore recommend that you try and create a fall back in case you are not able to reach our system to retrieve parcel rates.

As an example, we recommend referring to simplified shipping rates in case your system is unable to retrieve parcel rates from us.

Remember to set your system to automatically send Error reports to us so that we can fix problems in a timely manner.

Increasing response times when requesting shipping rates

Everyone loves fast responses. Here is a list of things you can implement to get faster responses from Fraktjakt.

One method to increase response times is to configure the parcel module to only retrieve and display a minimum amount of shippers. The less data communication (retrieval requests) the quicker the response times.

Our system is also able to provide parcel rates for those e-commerce platforms that require parcel rates before a customer indicates the destination of which the product will be sent (such as filling out their postal address during purchase). If you set your company’s address as the receiving address we are able to indicate a parcel rate before the purchase and provide an adjusted rate when customers fill in their address during purchase.

If parcel prices are not important however whether or not a certain parcel service is available to a certain address or not, you should send a *<no_prices>* element with the search.

If you are only going to use one shipping method, for example, if you use an AJAX query to update prices on a specific shipping method, then you should set your parcel module not to search for other shipping methods. Send instead shipping_product_id in the search xml.

If you think that providing or showing shipping agents (places where the customer can pick up their parcel shipments) then you should send the *<no_agents>* element. It quickens up the response times.

Other integration possibilities

If you like, you can let your users save their Fraktjakt user names and passwords in your application. This will enable you to open an external browser window or an *iFrame* using the following URL:s:

You can add the format-parameter "redirect" to the URL:s

- *redirect=ecommerce* is a reserved parameter that you may optionally include in order to eliminate layout information from the Fraktjakt page that your user sees (in case you want to apply a different layout locally). Suitable when used in an iFrame.
- *redirect=clean* removes our wrapper with sidehead, side footer and menu and replaces them with "Powered by Fraktjakt" at the bottom. Suitable in an iFrame which you want to easily layout to fit the webshop.
- *redirect=raw* removes all layout and returns parts of a webpage.

Log in to Fraktjakt account

https://www.fraktjakt.se/account/login?login=USERNAME&password=PASSWORD&redirect=ecommerce

Change password

https://www.fraktjakt.se/account/change_email?redirect=ecommerce

Administration of the webshop associated with the logged-in user

https://www.fraktjakt.se/webshops/change?redirect=ecommerce

Administration of addresses

https://www.fraktjakt.se/account/change_addresses?redirect=ecommerce

Link to a frieght-query

https://www.fraktjakt.se/fraktjakt/result?&number_parcels[id]=1&country_from[id]=2&country_to[id]=2&referrer_code=647fdd28c4518861daef7a52aeeac6c5b2ab7391&vikt[0]=4.0&langd[0]=40.0&bredd[0]=40.0&hojd[0]=40.0&postal_code_from=55321

Track a shipment

If you want to provide your users with a link to track their shipment, send them this link:
https://www.fraktjakt.se/trace/list_shipment/SHIPMENT_ID?redirect=ecommerce

SHIPMENT_ID is the one sent back in the reply from Order API when the order is placed in

Fraktjakt. Not the one sent from the *query_api*.
You can also use the *track-API*.

Track a parcel

There is a way to get track 'n trace-info about a parcel whether it is bought in Fraktjakt or not. You need the parcel-id, tracking-number or similar and the shipping-company's number in fraktjakt.

https://www.fraktjakt.se/trace/create/?parcel[package_number]=1234&shipper[id]=4

parcel[package_number] is the number off the parcel you are tracking.
shipper[id] is the shipping-company's number in Fraktjakt.
It can be found in the result from the link to display what is currently sold in Fraktjakt. The most common ones are:

Bussgods – 4, DHL – 5, FedEx – 1015, Posten Logistik – 1, Schenker – 2, Privpak – 21, UPS – 1036

Information about what can be bought in Fraktjakt right now

This URL will generate an XML that shows what products are sold right now in Fraktjakt. It also has fields for description, terms and other things of interest for the customer.

https://www.fraktjakt.se/shipping_products/xml_list

When using Order API, call type 2 this call will tell you which shipping_product_id are available to use.

Notice! This is a list of all shipping services currently available for purchase in Fraktjakt. Before you use a service in Order Api 2 you have to make sure the service ships with given shipping addresses and that other product terms are met.

More about search results

By copying the shipping product ID that you receive in the XML from Fraktjakt's API, you can generate links to more information about the shipping product.

```
     - <shipping_product>

         <id>NR</id>
```

Terms and conditions for the shipping product

https://www.fraktjakt.se/shipping_products/show_terms/**NR**

Other information about the shipping product

https://www.fraktjakt.se/shipping_products/show/**NR**

Support

Contact us https://www.fraktjakt.se/om_fraktjakt/kontakt

Fraktjakt AB
Gjuterigatan 9
55318 Jönköping
SWEDEN

VAT number: SE556751901101

E-post: info@fraktjakt.se
Web: https://www.fraktjakt.se/
Webchat: https://www.fraktjakt.se/om_fraktjakt/kontakt

Customer support office hours:
Monday – Friday: 8.00 – 16.00 CET (Central European Time)
Saturday and Sunday: Closed
Swedish national holidays: Closed

Appendix 1 - Language

The following languages are now supported in Fraktjakt:

Code	Language
sv	Swedish
en	English

The above language codes follow ISO 639-1. Se details in Wikipedia: https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.

<language> elements are included in two parts of the XML for API calls:

- in the <consignor> element, so Fraktjakt knows which language to respond to the sender of the API call (identified directly or indirectly by the consignor block);
- in <address_to> elements, so Fraktjakt knows which language to use when communicating with the recipients of shipments (normally customers of the consignor).

Support for English is currently only partial in Fraktjakt's responses to API calls, but it is complete in all correspondence with the receivers of shipments. If <sender_email> elements are included in Order-API calls, the senders of the shipments also receive all of their communication in the specified language.

Fraktjakt's branding and customization features are fully functional in both Swedish and English

Appendix 2 – Countries in Fraktjakt

The following countries can be selected in Fraktjakt API calls by submitting their codes (as shown in the table) in the appropriate address elements (See Address element). Note, however, that most shipping services do not serve all of these countries.

Fraktjakt can be used for shipping to or from Sweden, but either the originating address (<address_from>), the destination address (<address_to>), or both must be in Sweden.

The country codes follow ISO 3166-1 alpha-2 (See https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2).

Name	Code	Has Postal Codes
Aaland Islands	AX	YES
Afghanistan	AF	YES
Albania	AL	YES
Algeria	DZ	YES
American Samoa	AS	YES
Andorra	AD	YES
Angola	AO	YES
Anguilla	AI	YES
Antarctica	AQ	NO
Antigua and Barbuda	AG	NO
Argentina	AR	YES
Armenia	AM	YES
Aruba	AW	NO
Ascension Island	AC	YES
Australia	AU	YES
Austria	AT	YES
Azerbaijan	AZ	YES
Azores (Portuguese autonomous region)	PT	YES
Bahamas	BS	NO
Bahrain	BH	YES
Bangladesh	BD	YES
Barbados	BB	YES
Belarus	BY	YES
Belgium	BE	YES

Name	Code	Has Postal Codes
Belize	BZ	NO
Benin	BJ	NO
Bermuda	BM	YES
Bhutan	BT	YES
Bolivia	BO	NO
Bosnia and Herzegovina	BA	YES
Botswana	BW	NO
Bouvet Island	BV	NO
Brazil	BR	YES
British Indian Ocean Territory	IO	YES
Brunei Darussalam	BN	YES
Bulgaria	BG	YES
Burkina Faso	BF	NO
Burundi	BI	NO
Cambodia	KH	YES
Cameroon	CM	NO
Canada	CA	YES
Canary Islands	IC	NO
Cape Verde	CV	YES
Cayman Islands	KY	YES
Central African Republic	CF	NO
Ceuta and Melilla	EA	NO
Chad	TD	YES
Chile	CL	YES
China	CN	YES
Christmas Island	CX	YES
Clipperton Island	CP	NO
Cocos (Keeling) Islands	CC	YES
Colombia	CO	YES
Comoros	KM	NO
Congo	CG	NO
Congo, Democratic Republic of the	CD	YES
Cook Islands	CK	NO
Costa Rica	CR	YES
Croatia	HR	YES
Cuba	CU	YES
Curacao	CW	NO
Cyprus	CY	YES

Name	Code	Has Postal Codes
Czech Republic	CZ	YES
Denmark	DK	YES
Diego Garcia	DG	NO
Djibouti	DJ	NO
Dominica	DM	NO
Dominican Republic	DO	YES
Ecuador	EC	YES
Egypt	EG	YES
El Salvador	SV	YES
Equatorial Guinea	GQ	NO
Eritrea	ER	NO
Estonia	EE	YES
Ethiopia	ET	YES
Falkland Islands (Malvinas)	FK	YES
Faroe Islands	FO	YES
Fiji	FJ	NO
Finland	FI	YES
France	FR	YES
France, Metropolitan	FX	NO
French Guiana	GF	YES
French Polynesia	PF	YES
French Southern Territories	TF	NO
Gabon	GA	YES
Gambia	GM	NO
Georgia	GE	YES
Germany	DE	YES
Ghana	GH	NO
Gibraltar	GI	YES
Greece	GR	YES
Greenland	GL	YES
Grenada	GD	NO
Guadeloupe	GP	YES
Guam	GU	YES
Guatemala	GT	YES
Guernsey	GG	YES
Guinea	GN	YES
Guinea-Bissau	GW	YES
Guyana	GY	NO

Name	**Code**	**Has Postal Codes**
Haiti	HT	YES
Heard Island and McDonald Islands	HM	YES
Honduras	HN	YES
Hong Kong	HK	NO
Hungary	HU	YES
Iceland	IS	YES
India	IN	YES
Indonesia	ID	YES
Iran, Islamic Republic of	IR	YES
Iraq	IQ	YES
Ireland	IE	NO
Isle of Man	IM	YES
Israel	IL	YES
Italy	IT	YES
Ivory Coast	CI	NO
Jamaica	JM	YES
Japan	JP	YES
Jersey	JE	YES
Jordan	JO	YES
Kazakhstan	KZ	YES
Kenya	KE	YES
Kiribati	KI	NO
Korea, Democratic People's Republic of	KP	NO
Korea, Republic of	KR	YES
Kosovo (use Serbia)	XK	YES
Kuwait	KW	YES
Kyrgyzstan	KG	YES
Lao People's Democratic Republic	LA	YES
Latvia	LV	YES
Lebanon	LB	YES
Lesotho	LS	YES
Liberia	LR	YES
Libyan Arab Jamahiriya	LY	YES
Liechtenstein	LI	YES
Lithuania	LT	YES
Luxembourg	LU	YES
Macao	MO	NO
Macedonia	MK	YES

Name	Code	Has Postal Codes
Madagascar	MG	YES
Malawi	MW	NO
Malaysia	MY	YES
Maldives	MV	YES
Mali	ML	NO
Malta	MT	YES
Marshall Islands	MH	YES
Martinique	MQ	YES
Mauritania	MR	NO
Mauritius	MU	NO
Mexico	MX	YES
Micronesia, Federated States of	FM	YES
Moldova	MD	YES
Monaco	MC	YES
Mongolia	MN	YES
Montenegro	ME	YES
Montserrat	MS	YES
Morocco	MA	YES
Mozambique	MZ	YES
Myanmar	MM	YES
Namibia	NA	YES
Nauru	NR	NO
Nepal	NP	YES
Netherlands	NL	YES
Netherlands Antilles	AN	NO
New Caledonia	NC	YES
New Zealand	NZ	YES
Nicaragua	NI	YES
Niger	NE	YES
Nigeria	NG	YES
Niue	NU	NO
Norfolk Island	NF	YES
Northern Mariana Islands	MP	YES
Norway	NO	YES
Oman	OM	YES
Pakistan	PK	YES
Palau	PW	YES
Palestinian National Authority	PS	NO

Name	Code	Has Postal Codes
Panama	PA	YES
Papua New Guinea	PG	YES
Paraguay	PY	YES
Peru	PE	YES
Philippines	PH	YES
Pitcairn	PN	YES
Poland	PL	YES
Portugal	PT	NO
Puerto Rico	PR	YES
Qatar	QA	NO
Réunion	RE	YES
Romania	RO	YES
Russia	RU	YES
Rwanda	RW	NO
Saint Barthélemy	BL	YES
Saint Helena	SH	YES
Saint Kitts and Nevis	KN	NO
Saint Lucia	LC	YES
Saint Martin (French part)	MF	YES
Saint Pierre and Miquelon	PM	YES
Saint Vincent and the Grenadines	VC	YES
Samoa	WS	NO
San Marino	SM	YES
Sao Tome and Principe	ST	NO
Saudi Arabia	SA	YES
Senegal	SN	YES
Serbia	RS	YES
Serbia and Montenegro (ended 2006)	CS	NO
Seychelles	SC	NO
Sierra Leone	SL	NO
Singapore	SG	YES
Slovakia	SK	YES
Slovenia	SI	YES
Solomon Islands	SB	NO
Somalia	SO	NO
South Africa	ZA	YES
South Georgia and the South Sandwich Islands	GS	YES

Name	**Code**	**Has Postal Codes**
Spain	ES	YES
Sri Lanka	LK	YES
Sudan	SD	YES
Suriname	SR	NO
Svalbard and Jan Mayen	SJ	YES
Swaziland	SZ	YES
Sweden	SE	YES
Switzerland	CH	YES
Syrian Arab Republic	SY	NO
Taiwan, Province of China	TW	NO
Tajikistan	TJ	YES
Tanzania, United Republic of	TZ	NO
Thailand	TH	YES
Timor-Leste	TL	NO
Togo	TG	NO
Tokelau	TK	NO
Tonga	TO	NO
Trinidad and Tobago	TT	YES
Tristan da Cunha	TA	NO
Tunisia	TN	YES
Turkey	TR	YES
Turkmenistan	TM	YES
Turks and Caicos Islands	TC	YES
Tuvalu	TV	NO
Uganda	UG	NO
Ukraine	UA	YES
United Arab Emirates	AE	NO
United Kingdom	GB	YES
United States	US	YES
United States Minor Outlying Islands	UM	NO
Uruguay	UY	YES
Uzbekistan	UZ	YES
Vanuatu	VU	NO
Vatican City State	VA	YES
Venezuela	VE	YES
Viet Nam	VN	YES
Virgin Islands, British	VG	YES
Virgin Islands, U.S.	VI	YES

Name	**Code**	**Has Postal Codes**
Wallis and Futuna	WF	YES
Western Sahara	EH	NO
Yemen	YE	NO
Zambia	ZM	YES
Zimbabwe	ZW	NO