Fraktjakt API

Version 1.16 EN   2011-02-20

Contents

Welcome!

This document describes how you can integrate Fraktjakt.se with other services, such as web shops. The Fraktjakt API (application programming interface) can be used by business systems, e-commerce systems, on-line marketplaces, etc. to communicate with Fraktjakt.se.

Here's an example of what a call to the Fraktjakt API could look like:

http://www.fraktjakt.se:3333/catalog/index.php?
cPath=1&osCsid=c2nbmih9r1rnb5tkv4t8vfd3p5

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

If you are integrating Fraktjakt.se with a web shop built on OsCommerce, we recommend that you use the Fraktjakt OsCommerce Module for this instead of doing an integration from scratch. Download the Fraktjakt OsCommerce Module and its documentation from http://www.fraktjakt.se/services/download/. It's free to use.

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

You can use this link to contact us:

http://www.fraktjakt.se/om_fraktjakt/kontakt

Latest version of the documentation describing how to manage a webshop in Fraktjakt can be found at:

http://www.fraktjakt.se/downloads/fraktjakt_manual_webbutik.pdf

Latest version of this documentation can be found at:

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

General

Fraktjakt is a service for comparing and purchasing shipping services over the Internet. It is available both as a web site (http://www.fraktjakt.se) and as an API that can be called by other services on the Internet. This document describes how to use the API.

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

In the current version, the API is solely XML-based. We will add interfaces based on Web Services or SOAP as soon as we see that there is a demand for this.

Versions

Important changes made in this document

Version	Date	Change and additions
1.16	110220	New link how to track parcels not from Fraktjakt. Anonymous sender is now possible on the shipping-documents.
1.15	101228	Trace-api to keep track on your shipments
1.14	101118	Better explanation for how to add commodities to a package to get the right measurements.
1.13	101021	Exemplification on how to build a parcel
1.12	101020	More info about how to make an integration.
1.11	100921	Web shops 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 web shop, Fraktjakt gives you **simpler shipping support** 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 shipping contracts, since we have already negotiated good shipping contracts for you. Fraktjakt also makes your shipping management very simple, so you can focus more on selling your products.

We help you save on both domestic and international shipping. This is possible because we have volume based contracts with the shippers. You get access to these discounted prices when you use Fraktjakt with your web shop.

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 web shop;
- 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.

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.

The Servers

Fraktjakt has two separate server environments that you can use when you integrate.

Test environment – The test environment is a sandbox for development and testing. Here you can do as many API calls as you want without causing any real shipments to be ordered, and thus without incurring any costs. You cannot create real shipping documents here, either.

The test environment may show shipping alternatives that aren't actually purchaseable via Fraktjakt.

Remember that the transactions that are performed in the test environment are not saved for any long periods of time. We purge the test server now and then to keep it efficient.

> http://api2.fraktjakt.se

Production environment – This is where you will send your API calls when you have finished testing and are using the Fraktjakt API live to purchase real shipping services. In contrast to the test environment, this environment only shows shipping services that can be purchased via Fraktjakt.

> http://api1.fraktjakt.se

Note that this is not the URL that vistors to the Fraktjakt.se web site use. Fraktjakt uses separate servers in order to provide more efficient searches and better stability and security for your web shop.

Newer use http://www.fraktjakt.se in your calls. That will seriously slow down your application, probably give you the wrong result and make bugg-searching much harder for us.
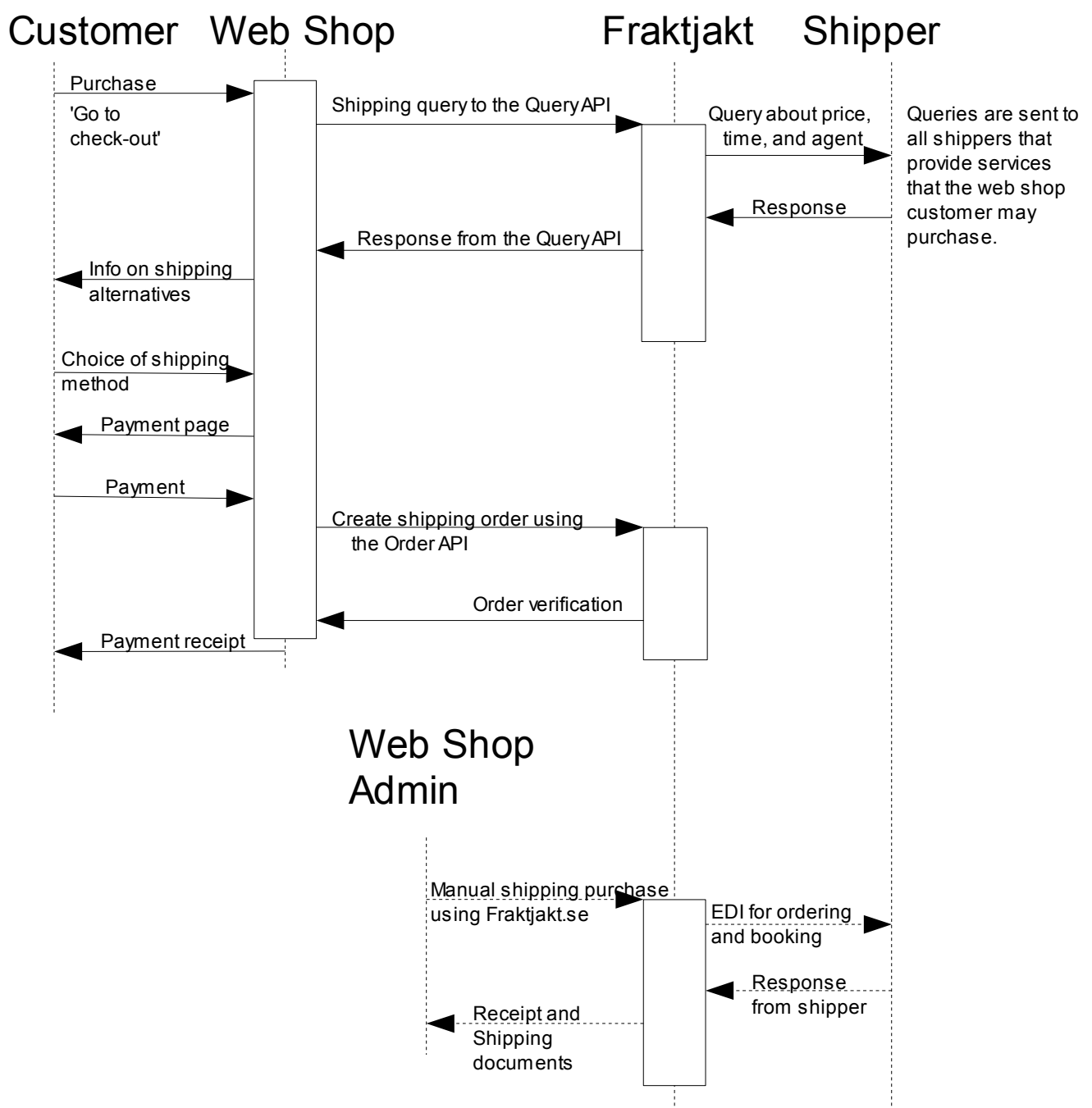
Of course, even though the servers are separate, they work with the same data. Thus, you will have no problem browsing to http://www.fraktjakt.se to manage your shipments and print out your shipping documents.

Interaction Diagram

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

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

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

Customer	Web Shop		Fraktjakt	Shipper

Purchase
'Go to check-out'

Shipping query to the Query API

Query about price, time, and agent

Queries are sent to all shippers that provide services that the web shop customer may purchase.

Response

Response from the Query API

Info on shipping alternatives

Choice of shipping method

Payment page

Payment

Create shipping order using the Order API

Order verification

Payment receipt

Web Shop Admin

Manual shipping purchase using Fraktjakt.se

EDI for ordering and booking

Response from shipper

Receipt and Shipping documents

Creation of a web shop account at Fraktjakt.se

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

These are created when you follow the following steps to register a web shop account:

1. To create a web shop account in Fraktjakt's test environment, go to the page titled "**Registrera en webbutik**": http://api2.fraktjakt.se/account/register

   [Later on, when you have completed integrating and testing, you will need to create a new web shop account in the production environment using the following link: http://api1.fraktjakt.se/account/register.]

2. Fill in the entire form and then press the button labeled "Registrera" at the bottom of the page. If you have filled in the form correctly, an activation code will be sent to the e-mail address that you entered on this page.

3. Enter the activation code that was sent to you by e-mail after the previous step.

4. If you didn't enter a delivery address ("godsadress") or billing address ("faktureringsadress") in step 1, above, you need to do that now. Go to the tab named "Adresser" and fill in your address information.

5. Go to the page titled "**Webshop**": http://api2.fraktjakt.se/webshops/change

   Your **Consignor ID** och **Consignor key** are displayed on this page.

   [When you are ready to go live, you will use the following link instead: http://api1.fraktjakt.se/webshops/change.]

General information about Fraktjakt's XML-based API

The API supports both UTF-8 och 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 tends 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 documentationen 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. http://en.wikipedia.org/wiki/Query_string#URL_encoding

The following applies to all calls made to Fraktjakt via the XML-based API:

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 http://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. The XML data that your system sends to Fraktjakt should include an element named "value". This is the value of what is being shipped (e.g. the value of a purchase that was done in a web shop). Although this element is optional, we recommend that you include it; some shipping prices and shipping services cannot be accessed without it.

4. The XML data that your system sends to Fraktjakt must include an element named "consignor". This element, in turn, contains the following elements:
   - id – The consignor ID that is shown on your configuration page at Fraktjakt.se. See step 5 in the section of this document titled *Creation of a web shop account at Fraktjakt.se* (above).

   - key – The consignor key that is shown on your configuration page at Fraktjakt.se. See step 5 in the section of this document titled *Creation of a web shop account at Fraktjakt.se* (above).

   - currency – the currency for the purchase. Use ISO 4217 notation. Currently, only "SEK" is supported. [This element is optional.]

   - language – the language that Fraktjakt is to use when responding to this API call. Use ISO 639-1. Currently only "sv" is supported (Swedish). [This element is optional.]

   - encoding – the encoding that Fraktjakt is to use when responding to this API call. We recommend UTF-8, although ISO-8859-1 is also supported. If this sub-element is omitted, Fraktjakt's response will use UTF-8 encoding. [This element is optional.]

     Remember to URL-encode the XML-string - http://en.wikipedia.org/wiki/Query_string#URL_encoding

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 www.fraktjakt.se; however, the Query API is faster and provides results that are fine tuned for web shops or other types of e-commerce systems.

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 web shop configuration page at Fraktjakt.se.

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

The Query API uses the following URI's:
Test environment:	http://api2.fraktjakt.se/fraktjakt/query_xml
Production environment:	http://api1.fraktjakt.se/fraktjakt/query_xml

Newer use http://www.fraktjakt.se. That will seriously slow down your application, probably give you the wrong result and make bugg-searching much harder for us.

XML element ("tag") and example of contents	Description
<?xml version="1.0" encoding="iso-8859-1"?>	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</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_guarantie>`**[0 | 1]**`</time_guarantie>`	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)
- `<consignor>`	Information about the consignor, i.e. the web shop or other entity that is performing the query.
`<id>`**10351**`</id>`	The consignor's customer ID in Fraktjakt. Data type: INT Required: Yes
`<key>`**gh50311h3223j3j3908dsjo3**`</key>`	The consignor's login key in Fraktjakt. Data type: STRING Max chars: 64 Required: Yes
`<currency>`**SEK**`</currency>`	Currency in ISO 4217 notation. Data type: STRING Max chars: 3 Required: No
`<language>`**sv**`</language>`	Language in ISO 639-1 notation (6) Data type: STRING Max chars: 2 Required: No Default: sv
`<encoding>`[**UTF-8	ISO-8859-1**]`</encoding>`	The encoding that applies to text in the XML data. Data type: STRING Required: No Default: UTF-8
`</consignor>`	
- `<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. Minimum count = 1
- `<parcel>`	Data pertaining to a parcel.
`<weight>`**3.2**`</weight>` (OBS! siffrorna skall ha decimal-punkt)	The parcel's weight in kg. Data type: FLOAT Required: Yes
`<length>`**35**`</length>`	The parcel's length in cm.

(OBS! siffrorna skall ha decimal-punkt)	The length is the longest side of the parcel. Data type: FLOAT Required: No
<width>**23.5**</width> (OBS! siffrorna skall ha decimal-punkt)	The parcel's width in cm. The width is the second longest side of the parcel. Data type: FLOAT Required: No
<height>**15**</height> (OBS! siffrorna skall ha decimal-punkt)	The parcel's height in cm. Data type: FLOAT Required: No
</parcel>	
</parcels>	
- <address>	The address to which the shipment is to be sent. Alternate name: address_to Required: no – but if it is not used, the shipping address for the web-shop will be used. That will give a quick, but rather incorrect query.
<street_address_1>**Gjuterig. 9**</street_address_1>	The first line of the street address. Data type: STRING Max chars: 255 Required: Yes
<street_address_2>**Vån 7 C**</street_address_2>	The second line of the street address. Data type: STRING Max chars: 255 Required: No
<postal_code>**55318**</postal_code>	The postal code. Data type: STRING Max chars: 16 Required: Yes – except in countries that don't use postal codes, e.g. Gambia.
<city_name>**Jönköping**</city_name>	The city. Data type: STRING Max chars: 32 Required: No, if in Sweden and postal_code are provided.
<residential>**[0 | 1]**</residential>	Flag to indicate whether the address is a residential address. Data type: BOOLEAN Required: No Default = 1 (residential)
<country_code>**SE**</country_code>	Country code in ISO 3166-1 alpha-2 notation (4) (5) Data type: STRING Max chars: 2 Required: Yes
<country_subdivision_code>**F**</country_subdivision_code>	Country subdivision code Data type: STRING Max chars: 3 Required: No (1)
</address>	
- <address_from>	OPTIONAL! The address from which

	the shipment is to be sent. If this element is omitted, the shipping address that you have entered on your configuration page at Fraktjakt.se will be used. Omitting this element will improve efficiency and give some extra functionality. Required: No (2)
`<street_address_1>`**Storg 1**`</street_address_1>`	The first line of the street address. Data type: STRING Max chars: 255 Required: Yes
`<street_address_2></street_address_2>`	The second line of the street address. Data type: STRING Max chars: 255 Required: No
`<postal_code>`**12345**`</postal_code>`	The postal code. Data type: STRING Max chars: 16 Required: Yes – except in countries that don't use postal codes, e.g. Gambia.
`<city_name>`**Stockholm**`</city_name>`	The city. Data type: STRING Max chars: 32 Required: No, if in Sweden and postal_code are provided.
`<residential>`**[0 | 1]**`</residential>`	Flag to indicate whether the address is a residential address. Data type: BOOLEAN Required: No Default = 1 (residential)
`<country_code>`**SE**`</country_code>`	Country code in ISO 3166-1 alpha-2 notation (4) (5) Data type: STRING Max chars: 2 Required: Yes
`<country_subdivision_code>`**F**`</country_subdivision_code>`	Country subdivision code Data type: STRING Max chars: 3 Required: No (1)
`</address_from>`	
`</shipment>`	

Notes:
 1. If the shipment is destined for the United States, Puerto Rico, Canada, or related areas, a <country_subdivision_code> must be included in the address.

2. <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 web shop 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.

3. 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.

4. http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

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

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

Example off a working xml

This Xml should return a result if you just change the consignor id and key to your own.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<shipment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <value>350</value>
  <consignor>
    <id>1234</id>
    <key>SECRET CODE</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>UTF-8</encoding>
  </consignor>
  <no_agents>1</no_agents>
  <parcels>
    <parcel>
      <weight>1.05</weight>
    </parcel>
    <parcel>
      <weight>4</weight>
      <length>30</length>
      <width>20</width>
      <height>10</height>
    </parcel>
  </parcels>
  <address>
    <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>
</shipment>
```

Quick queries

It is possible to do a quick and simple query that relies on cached data in the Fraktjakt servers. This can be useful in situations where speed is more important than accuracy, such as when you want to quickly present an approximate price to your customers, but don't need to give them the final price yet. It is also useful in situations where you don't know exactly where a package is going to be sent.

All you need to do is to include an exact duplicate of the contents of <address> in <address_to>. This will cause Fraktjakt to do a quick query based heavily on cached data.
[NB: The next version of the Query API will include a simpler mechanism for requesting a quick query.]

IMPORTANT! DO NOT CREATE AN ORDER USING THE RESULTS OF A QUICK QUERY. Instead, do a new query when it is time to create the actual order. If you do that the shipper will not get the right address-information and the shipment will be shipped to the wrong address. You need to first create a new shipment with the right information. It is then advisable to use the shipping_product_id you are interested in buying.

XML for Query API responses

XML element ("tag") and example of contents	Description
<?xml version="1.0" encoding="iso-8859-1"?>	
- <shipment>	The shipment element is the containing element for all query results.
<id>**67887**</id>	Fraktjakt's shipment ID. This is used in all future references to this particular query result. Data type: INTEGER Required: Yes
<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
- <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.
<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>**En till tre dagar, före kl 10. Garanterat**</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>` (OBS! siffrorna kommer att ha decimal-punkt)	The shipping product's total price. Data type: FLOAT Required: Yes
`<tax_class>`25.00`</tax_class>` (OBS! siffrorna kommer att ha decimal-punkt)	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>`**http://api.fraktjakt.se/agents/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>`**http://api.fraktjakt.se/agents/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
`</shipping_product>`	
`</shipping_products>`	
`</shipment>`	

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.

Requery API

It is also possible to get a cached query result by calling the Requery API.
http://api2.fraktjakt.s/fraktjakt/requery_xml   (test environment)
http://api1.fraktjakt.s/fraktjakt/requery_xml   (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.

Newer use http://www.fraktjakt.se. That will seriously slow down your application, probably give you the wrong result and make bugg-searching much harder for us.

Example:

```
<?xml version=\"1.0\" encoding=\"iso-8859-1\"?>
<shipment>
  <value>10</value>
  <consignor>
    <id>1212</id>
    <key>1212121212121212121212121212121212</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>ISO-8859-1</encoding>
  </consignor>
  <shipment_id>67887</shipment_id>
</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 web shop platforms, such as OsCommerce, the shipping is purchased immediately after a web shop customer completes payment for items that the customer is purchasing in the web shop. However, other web shop 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:

 http://api2.fraktjakt.se/orders/order_xml   (test)

 http://api1.fraktjakt.se/orders/order_xml   (live / produktion)

Newer use http://www.fraktjakt.se. That will seriously slow down your application, probably give you the wrong result and make bugg-searching much harder for us.

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.

XML for Order API calls

XML element ("tag") and example of contents	Description
<?xml version="1.0" encoding="iso-8859-1"?>	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 web shop or other entity that is sending the order.
<id>10351</id>	The consignor's customer ID in Fraktjakt.

	Data type: INT Required: Yes
`<key>`**`gh5031113223j3j3908dsjo3`**`</key>`	The consignor's login key in Fraktjakt. Data type: STRING Max chars: 64 Required: Yes
`<currency>`**`SEK`**`</currency>`	Currency in ISO 4217 notation. Data type: STRING Max chars: 3 Required: No
`<language>`**`sv`**`</language>`	Language in ISO 639-1 notation (6) Data type: STRING Max chars: 2 Required: No Default: sv
`<encoding>`[**`UTF-8 | ISO-8859-1`**]`</encoding>`	The encoding that applies to text in the XML data. Data type: STRING Required: No Default: UTF-8
`</consignor>`	
`<referred_consignor>`	Use this tag if it isn't the consignor who will finish and pay for the transportation. It will then refer to the user who will pay for the freight. The answer 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>`1212121212121212121212121`</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>`
 	
`Ska vara ett av de id som erhölls tidigare från Sök-API'et query_xml`
	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	Used if you want to put another

`name</anonymous_sender>`	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
`<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**
`<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 https://tid.tullverket.se/taric/TAG_Search.aspx	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>` Tillåten värden: EA = styck DZ = dussin L = liter ML = mililiter KG = kilogram Default är EA (each) om inget värde anges.	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.**
`<description>`**15**`</description>` (OBS! Minst 15 tecken)	Description of the commodity type. Data type: STRING Min tecken: 15 Max chars: 128 Required: **Yes when the shipment is international**
`<country_of_manufacture>`**US**`</country_of_ma nufacture>`	Landskod, enligt ISO 3166-1 alpha-2 (5) standarden. Data type: STRING Max chars: 2

`Default är SE om inget värde anges.`	Required: No, **men kan anges i frakt till utlandet**
`<weight>`**13.1**`</weight>` (OBS! siffrorna **skall** ha decimal-punkt)	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>` (OBS! siffrorna skall ha decimal-punkt)	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>`	
- `<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>`**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: No
`</recipient>`	
`<sender_email>`info@exempel.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 the tag sender_email_link in the reply and an email to that address when the order is paid. (7) Data type: STRING Max chars: 64 Required: No
`<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 web shops configuration page at Fraktjakt.se. Required: No
`<driving_instruction>Hit och dit</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>Dörrkod 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>2011-12-10</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. http://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 http://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", which can be found under the tab labeled "Mina Sidor".

Example of a working XML

This Xml should return a result if you just change the consignor id and key to your own. And takes a shipment_id från one of your query-results.

```xml
<?xml version=\"1.0\" encoding=\"iso-8859-1\"?>
<OrderSpecification>
  <consignor>
    <id>2</id>
    <key>eb412ec51b39e918b203c6aa3e48c0cd4d36e650</key>
    <currency>SEK</currency>
    <language>sv</language>
    <encoding>utf-8</encoding>
  </consignor>
  <referred_consignor>
    <id>1234</id>
    <key>SECRET_CODE</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´</company_to>
    <telephone_to>036121210</telephone_to>
  </recipient>
  <booking>
    <pickup_date>2011-12-10</pickup_date>
    <driving_instruction>Upp för backen och sedan över ån, in i
kammarn.</driving_instruction>
  <user_notes>Dörrkod 1112</user_notes>
  </booking>
</OrderSpecification>
```

Answere

XML element ("tag") and example of contents	Description
`<?xml version="1.0" encoding="iso-8859-1"?>`	
`- <result>`	
`<order_id>67887</order_id>`	Fraktjakt's order ID Data type: INT Required: Yes
`<shipment_id>45654</shipment_id>`	Fraktjakts shipment ID (1) Data type: INT Required: Yes
`<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
`<currency>SEK</currency>`	Currency. Data type: STRING Max chars: 3 Required: Yes
`<payment_link>www.fraktjakt.se/orders/finish/1631?code=b6dfc12fc04ec98132da2eb1c1739272cc646ed9</payment_link>`	This tag contains a link to the payment page If referred_consignor-tag are given for paying user. (2) Data type: STRING Required: No
`<sender_email_link>http://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 *not* necessarily the same shipment_id that was used to create the order, since that id can be reused and changed if the shipment is splitted into several new shipments.
This id can be used to track the shipment with this links -
http://www.fraktjakt.se/trace/list_shipment/shipment_id
http://api1.fraktjakt.se/trace/xml_trace/shipment_id
See also Track-api for more information about this.
2.   Don't apply to services that are paid at the shippers own page/site, such as Schenker Privatpaket.

Track-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 retur4ns the answere 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.

http://api1.fraktjakt.se/trace/xml_trace/SHIPMENT_ID (api1 is changed to api2 during development)

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 be in a different status. This is done for all products that are handled by an agent.

Answere

XML element ("tag") and example of contents	Description
`<?xml version="1.0" encoding="iso-8859-1"?>`	
`- <result>`	
`<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>Ej betald</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>`	

1. The meaning of the different numbers:

   0 – Handled by the sender

   1 – Sent

   2 – Delivered

   3 – Signed

   4 - Returned

Webshops API (create a user and a webshop)

It can be practical to automatic create a user-account and web shop-information in Fraktjakt if you want to integrate a web shop-platform with Fraktjakt. You can send in your users account-information to Fraktjakt and create accounts in both systems. It is relativly 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:

http://api2.fraktjakt.se/webshops/create_xml   (test)

http://api1.fraktjakt.se/webshops/create_xml   (live / produktion)

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 cpied 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 web shop or other entity that is sending the order.
`<id>`**10351**`</id>`	The consignor's customer ID in Fraktjakt. Data type: INT Required: Yes
`<key>`**gh5031l13223j3j3908dsjo3**`</key>`	The consignor's login key in Fraktjakt. Data type: STRING Max chars: 64 Required: Yes
`<currency>`**SEK**`</currency>`	Currency in ISO 4217 notation. Data type: STRING Max chars: 3 Required: No
`<language>`**sv**`</language>`	Language in ISO 639-1 notation (6) Data type: STRING Max chars: 2 Required: No Default: sv
`<encoding>`**[UTF-8 | ISO-8859-1]**`</encoding>`	The encoding that applies to text

	in the XML data. Data type: STRING Required: No Default: UTF-8
`</consignor>`	
`<email>walderik@yahoo.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>hemlig</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</name>`	Webshop name Data type: STRING Required: Yes
`<url>http://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>Kör först hit, sedan dit och då är du framme</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
`<street_address_1>`**Storg 1**`</street_address_1>`	The first line of the street address. Data type: STRING Max chars: 255 Required: Yes
`<street_address_2></street_address_2>`	The second line of the street address. Data type: STRING Max chars: 255 Required: No
`<postal_code>`**12345**`</postal_code>`	The postal code. Data type: STRING Max chars: 16 Required: Yes – except in countries that don't use postal codes, e.g. Gambia.
`<city_name>`**Stockholm**`</city_name>`	The city. Data type: STRING Max chars: 32 Required: No, if in Sweden and postal_code are provided.
`<residential>`**[0 | 1]**`</residential>`	Flag to indicate whether the address is a residential address. Data type: BOOLEAN Required: No Default = 1 (residential)
`<country_code>`**SE**`</country_code>`	Country code in ISO 3166-1 alpha-2 notation (4) (5) Data type: STRING Max chars: 2 Required: Yes
`<country_subdivision_code>`**F**`</country_subdivision_code>`	Country subdivision code Data type: STRING Max chars: 3 Required: No (1)
`</sender_address>`	
`<invoice_address>`ADDRESS-tags`</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>`ADDRESS-tags`</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>`	

Answer

XML element ("tag") and example of contents	Description
`<?xml version="1.0" encoding="iso-8859-1"?>`	
`- <result>`	
`<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>`	

Test

You many use the forms found at the following URL's are to test your XML documents:

Query API:	http://api2.fraktjakt.se/xml/query_test
Requery API:	http://api2.fraktjakt.se/xml/requery_test
Order API:	http://api2.fraktjakt.se/xml/order_test

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

To test an 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 "Sök" 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

- **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 web shop 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 http://api2.fraktjakt.se (test and demo) or http://api1.fraktjakt.se (live) 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 (http://api2.fraktjakt.se) and one for the production environment (http://api1.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 web shop 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 web shop. The second module (The Fraktdata module) allows orders to be sent to Fraktjakt.

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

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

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

- **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 (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.

Exempel:
Say that you have three products with the measurements 30x20x15, 40x15x15 och 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.

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

```php
<?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 answers 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>-tag 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>-tag. It quickens up the response times.

Other integration possibilities

If you like, you can let your users save their Fraktjakt usernames and passwords in your application. This will enable you to open an external browser window or an iFrame using the following 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).

Log in to Fraktjakt account

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

Change password

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

Administration of the webshop associated with the logged-in user

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

Administration of addresses

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

Link to a frieght-query

```
http://api1.fraktjakt.se/fraktjakt/result?
&number_parcels[id]=1&country_from[id]=2&country_to[id]=2&referrer_code=647fdd28c4518861
daef7a52aeeac6c5b2ab7391&vikt[0]=4.0&langd[0]=40.0&bredd[0]=40.0&hojd[0]=40.0&postal_cod
e_from=55321
```

Track a shipment

If you want to provide your users with a link to track their shipment, send them this link:
```
http://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.
```
http://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.

`http://www.fraktjakt.se/shipping_products/xml_list`

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

`http://www.fraktjakt.se/shipping_products/show_terms/`**NR**

Other information about the shipping product

`http://www.fraktjakt.se/shipping_products/show/`**NR**

Support

Contact us

Fraktjakt AB
Gjuterigatan 9

55318 Jönköping

SWEDEN

VAT number: SE556751901101

E-post: info@fraktjakt.se
Web: www.fraktjakt.se
Telephone: 076-589 65 68

Customer support office hours:
Monday - Friday: 8.00–16.00
Saturday and Sunday: Closed