

Hunter Express

REST API



[General Information](#)

[Authorisation](#)

[Developing your API client](#)

[Testing](#)

[Domain Models](#)

[Authentication Error \(AUTH_ERROR\)](#)

[API Error \(ERROR\)](#)

[Quote \(QUOTE\)](#)

[Tax \(TAX\)](#)

[Goods Type \(GOODS_TYPE\)](#)

[Goods \(GOODS\)](#)

[Service \(SERVICE\)](#)

[Stop \(STOP\)](#)

[Contact \(CONTACT\)](#)

[Time window \(TIMEWINDOW\)](#)

[Cost \(COST\)](#)

[Event \(EVENT\)](#)

[API Methods](#)

[1. Quote \(POST /quote/get-quote\)](#)

[2. Book Job \(POST /booking/book-job\)](#)

[3. Job Connote Scan \(GET /booking/get-job-connote-scan\)](#)

[4. Job POD \(GET /booking/get-job-pod\)](#)

[5. Job Statuses \(GET /booking/get-job-statuses\)](#)

[6. Services \(GET /resources/services\)](#)

[7. Locations \(GET /resources/get-locations\)](#)

[8. Goods Types \(GET /resources/get-goods-types\)](#)

[Code Examples](#)

[FAQ](#)

[APPENDIX A - ERROR CODES AND MESSAGES](#)

[APPENDIX B - COMMON REQUEST ERRORS](#)

General Information

The Hunter Express REST API provides programmatic access to the HX backend. The current implementation consists of booking-related services, price estimation and various endpoints supporting these services.

To get started, contact the Hunter Express team and ask for an authorization key which is needed to access the API.

Below you'll find detailed descriptions of the available services as well as [code examples](#).

Authorisation

Accessing the API is secured using **Basic Authentication** over **SSL**. You will receive an authentication key from the Hunter Express team when you request access to the API.

The API is stateless and the key is needed with every request. It's submitted by using the *HTTP Authorization* header.

Authorization: Basic [your-key-here]

The key is a *Base64 encoded* string which looks similar to this:

Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l

Unauthenticated API calls will return an [authentication error](#) which looks like this:

```
{
  "message": "Full authentication is required to access this resource",
  "status": "UNAUTHORIZED"
}
```

HTTP status code for this response is **401**.

Developing your API client

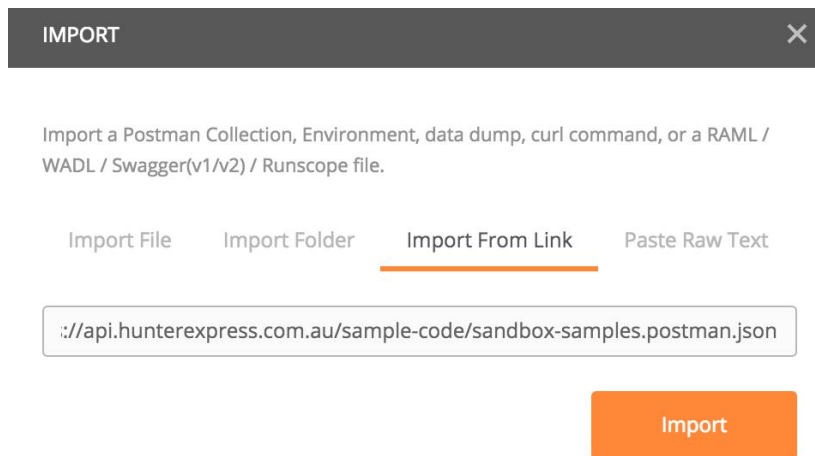
The API is a standard REST interface which means you can implement your API client using your preferred technology. As a reference, we have provided Java and PHP [code examples](#).

Chrome's Postman app is a handy tool to submit test queries against the API. You can [download](#) a predefined collection of queries and import it to Postman:

1. Install the Chrome Postman app.
2. Click the 'Import' button in the top left corner.






3. Either browse to the collection file (*sandbox-samples.postman.json*) which you downloaded or select the 'Import From Link' tab and paste the following URL into the field:





<https://api.hunterexpress.com.au/sample-code/sandbox-samples.postman.json>

4. Click 'Import' and you should now see the following API requests in the 'Collection' tab

History		Collections	
All	Me	Team	 
		HX REST API - Sandbox Samples 8 requests	
POST	/quote/get-quote		
POST	/booking/book-job		
GET	/booking/get-job-connote-scan		
GET	/booking/get-job-pod		
GET	/booking/get-job-statuses		
GET	/resources/get-services		
GET	/resources/get-locations		
GET	/resources/get-goods-types		

Notice the following key elements in the POST requests:

The **Headers** tab shows the *Content-Type* set to *application/json* and a correctly formatted *Authorization* header. This example value is a base64 encoded representation of the test credentials *hxws/hxws*.

POST 		https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/quote/get-quote			
Authorization		Headers (2)		Body 	
		Key		Value	
<input checked="" type="checkbox"/>	Content-Type			application/json	
<input checked="" type="checkbox"/>	Authorization			Basic aHh3czpoeHdz	
	New key			Value	

The **Body** tab contains the JSON payload which is transferred within the request body. Payload is posted as raw data (indicated by the selected radio button) and the highlighted *JSON* (*application/json*) shows that you have correctly set the *Content-Type* header.

POST ▾

https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/quote/get-quote

Authorization

Headers (2)

Body ●

Pre-request Script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON (application/json) ▾

1 ▾

{

2 "customerCode" : "DUMMY",

3 ▾ "fromLocation" : {

4 "suburbName" : "SYDNEY",

5 "postCode" : "2000",

6 "state" : "NSW"

7 },

8 ▾ "toLocation": {

9 "suburbName" : "BRISBANE",

10 "postCode" : "4000",

11 "state" : "QLD"

GET requests only need the *Authorization* header. You can inspect the supplied parameters by clicking the **Params** button next to the URL.

GET ▾

https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/get-job-connote-scan?custome...

Params

Authorization

Headers (1)

Body

Pre-request Script

Tests

Key	Value	Description
<input checked="" type="checkbox"/> Authorization	Basic aHh3czpoeHdz	
New key	Value	Description

HTTP Status Codes and Error Handling

The possible HTTP status codes which you can receive from the API are:

Code	Text	Description
200	OK	Returned when the request is successfully processed.
400	Bad Request	Returned when something is wrong with the request itself. This could be an incorrect endpoint or a missing/invalid request parameter. If this happens, first check the error description in the response body and then verify your request against the endpoint documentation.
401	Unauthorized	Returned if authenticating the request fails. See the Authorisation chapter for more details.
404	Not Found	The endpoint you requested was not found. Check that you have entered a valid endpoint.
415	Unsupported Media Type	Returned when the supplied 'Content-Type' HTTP header does not match the one expected by the endpoint.
500	Internal Server Error	Returned when a server side error occurs.

The first step when implementing your API client is to check the status code. Anything other than **200** indicates an error.

See [Appendix A](#) for error codes and descriptions and [Appendix B](#) for examples of a few common error scenarios.

Testing

The API has two different endpoints:

1) **Sandbox:** <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/>

All services starting with 'sandbox' are for testing. You can use the sandbox endpoint to develop your REST client and safely make test bookings.

The production credentials which you received from the HX team will not work with sandbox but instead you'll need to use these predefined test credentials:

Username: **hxws**

Password: **hxws**

Customer/account code: **DUMMY**

2) **Live/production:** <https://api.hunterexpress.com.au/live/rest/hxws/>

Once you have finished your testing and development phase you can switch to the live endpoint and start using your live/production credentials. Simply change the sandbox endpoints in your REST client to the production ones and you are ready to go live.

Username: **<supplied by HX>**

Password: **<supplied by HX>**

Customer/account code: **<your HX customer/account code>**

Domain Models

This is a list of domain objects representing the data returned by the endpoints.

Authentication Error (AUTH_ERROR)

Represents an authentication error.

Field name	Field type	Field description
message	String	The value is always: Bad credentials
status	String	The value is always: UNAUTHORIZED

Example:

```
{
  "message": "Bad credentials",
  "status": "UNAUTHORIZED"
}
```

API Error (ERROR)

Represents an API error.

Field name	Field type	Field description
webbitName	String	Internal error field.
errorCode	String	Error code.
errorMessage	String	Error message.
errorOccured	String	Server-side time when the error occurred. Helps to track down issues from the server-side log files. Formatted as ' yyyy-MM-dd HH:mm:ss '.

Example:

```
{
  "webbitName": "WS_getJobPOD",
  "errorCode": "WS_ERR_NOJOB",
}
```

```
"errorMessage": "Indicates that no job matching the specified criteria was found.",  
"errorOccured": "2017-06-23 23:36:31"  
}
```

Quote (QUOTE)

Contains the results of a request to quote the price of a delivery service.

Field name	Field type	Field description
serviceCode	String	Service code.
serviceName	String	Service name.
fee	Double	Quote fee, exclusive of any taxes
etd	String	Estimated time of delivery.
taxes	Array	List of TAX objects.

Tax (TAX)

Details of taxes and surcharges.

Field name	Field type	Field description
id	String	Identifies the type of tax or surcharge, e.g. "GST"
value	String	The taxes applicable to the "fee" field of a QUOTE

Example (QUOTE including TAX):

```
{  
  "serviceCode": "RF",  
  "serviceName": "Road Freight",  
  "fee": 42.40,  
  "etd": "Overnight",  
  "taxes": [  
    {  
      "id": "GST",  
      "value": 4.24  
    }  
  ]  
}
```

```
]
}
```

Goods Type (GOODS_TYPE)

Describes the type of goods being handled.

Field name	Field type	Field description
code	String	Goods type code. This is the 'typeCode' field in the GOODS object.
description	String	Description of the goods.

Example:

```
{
  "code": "AUTOM",
  "description": "Automotive/Car Part",
}
```

Goods (GOODS)

Describes a goods item

Field name	Field type	Field description
pieces	Integer	Number of items.
weight	Integer	Weight of the goods (kg).
width	Integer	Width (cm).
height	Integer	Height (cm).
depth	Integer	Depth (cm).
typeCode	String	A code that identifies the GOODS_TYPE of this item.

Example:

```
{
```

```
"pieces": 1,  
"weight": 2,  
"width": 20,  
"height": 30,  
"depth": 5,  
"typeCode": "ENV"  
}
```

Service (SERVICE)

Describes the type of delivery service.

Field name	Field type	Field description
code	String	Service code.
name	String	Service name.
primary	Boolean	Indicates whether or not this service can be used as a primary service.

Example:

```
{  
  "code": "AF",  
  "name": "Air Freight",  
  "primary": true  
}
```

Stop (STOP)

Describes a stop. Delivery jobs comprise two stops: one for pickup and one for delivery.

Field name	Field type	Field description
name	String	The name of the sender/receiver.*
suburbName	String	The name of the city/town/locality.

addressLine1	String	First line of the address of the sender/receiver.*
addressLine2	String	Second line of the address of the sender/receiver.*
stopReference	String	Optional customer-supplied.
state	String	The state or province abbreviation.
instructions	String	Optional instructions.*
contact	String	Details of the contact person at this stop. See CONTACT .
timeWindow	String	<p>The period of time during which the goods are to be picked up or delivered. See TIMEWINDOW.</p> <p>The time-windows are optional when the booking is made between 07:30 and 15:00 on business days, in which case the time-window will default to the current time. The same interval also applies to the 'earliest time' field. Otherwise, the time-window must be supplied for the stop 1.</p>

* Note: Although these fields accept any number of characters, only the first 30 will be preserved.

Example:

```
{
  "name": "James Doe",
  "suburbName": "BRISBANE",
  "addressLine1": "10 William St",
  "addressLine2": "Building 2",
  "postCode": "4000",
  "state": "QLD",
  "stopReference": "MY-REF-NO",
  "instructions": "Ask receptionist to call contact",
  "contact": {
    "name": "Jane Wonder",
    "phone": "82937823",
    "email": "jane@email.com"
  },
  "timeWindow": {
    "earliest": "2019-01-01 10:00",
    "latest": "2019-01-02 16:00"
  }
}
```

Contact (CONTACT)

Details of a contact person (or similar entity).

Field name	Field type	Field description
name	String	The name of the contact person.
phone	String	Phone number
email	String	Email address

Example:

```
{
  "name": "John Foster",
  "phone": "94386738",
  "email": "john.foster@email.com"
}
```

Time window (TIMEWINDOW)

Describes a time window used when booking a job

Field name	Field type	Field description
earliest	String	The start of the time window using the 24-hour format: yyyy-MM-dd HH:mm.
latest	String	The end of the time window using the 24-hour format: yyyy-MM-dd HH:mm.

Example:

```
{
  "earliest": "2019-01-01 10:00",
  "latest": "2019-01-02 16:00"
}
```

Cost (COST)

Describes the chargeable services provided (e.g. road freight, insurance), and the quantity on which the charge is based (e.g. kgs, value of goods).

Field name	Field type	Field description
serviceCode	String	RF
serviceName	String	Road Freight
quantity	Integer	2

Event (EVENT)

Describes a change of status in the delivery process, to allow the progress of the delivery to be tracked.

Field name	Field type	Field description
when	String	The date-time of the event
what	String	The type of event that occurred
who	String	Who (or what) actioned the event
info	String	Additional details

Example:

```
{  
  "when": "2017-06-27T09:15:00+10:00",  
  "what": "Delivered",  
  "who": "0697",  
  "info": "PoD: Fred"  
}
```


API Methods

These are the services available in the Hunter Express API.

1. Quote (POST /quote/get-quote)

Use this service to obtain quotes for your transportation requirements.

POST /quote/get-quote		
Request	Parameters	QUOTE_PARAMS
	Headers	TEST: Authorization:Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	QUOTE_RESULT

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/quote/get-quote>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/quote/get-quote>

Quote Params (QUOTE_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer.
fromLocation	LOCATION	yes	From-location.
toLocation	LOCATION	yes	To-location.
goods	List of GOODS	yes	A list of goods to be picked up and delivered.

Params example:

```
{
  "customerCode": "DUMMY",
  "fromLocation": {
    "suburbName": "SYDNEY",
```

```

    "postCode" : "2000",
    "state" : "NSW"
  },
  "toLocation": {
    "suburbName" : "BRISBANE",
    "postCode" : "4000",
    "state" : "QLD"
  },
  "goods" : [
    {
      "pieces" : "1",
      "weight" : "2",
      "width" : "20",
      "height" : "30",
      "depth" : "15",
      "typeCode" : "ENV"
    }
  ]
}

```

Quote Result (QUOTE_RESULT)

Field name	Field type	Field description
	List of QUOTE objects	List of quotes, one for each available service type.

Result example:

```

[
  {
    "serviceCode": "RF",
    "serviceName": "Road Freight",
    "fee": 42.40,
    "etd": "Overnight",
    "taxes": [
      {
        "id": "GST",
        "value": 4.24
      }
    ]
  },
  {
    "serviceCode": "AF",
    "serviceName": "Air Freight",
    "fee": 72.40,
    "etd": "Overnight",

```

```

    "taxes": [
      {
        "id": "GST",
        "value": 7.24
      }
    ]
  }
]

```

2. Book Job (POST /booking/book-job)

This service is used to create a delivery job.

POST /booking/book-job		
Request	Parameters	BOOKING_PARAMS
	Headers	<i>TEST:</i> Authorization: Basic aHh3czpoeHdz <i>PRODUCTION:</i> Authorization: Basic [your-key]
Response	Body	BOOKING_RESULT

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/book-job>
PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/booking/book-job>

Booking Params (BOOKING_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer.
reference1	String	no	Optional customer supplied reference.
reference2	String	no	Optional customer supplied. If this field is supplied, and if it conforms to the pattern for an HX connote number (i.e. [A-Z]{2}[0-9]{6}), then it will be treated as a connote number, and will be echoed as the "trackingNumber" in the call's response. Otherwise, it will be ignored, and replaced by a freshly generated connote number.
reference3	String	no	Optional customer supplied reference.

reference4	String	no	Optional customer supplied reference.
primaryService	String	yes	"Primary" service for the job (use the serviceCode returned by the quote or services endpoints).
secondaryServices	Array	no	A list of optional extra services (use the serviceCode in the extraServices listing of the quote).
connoteFormat	String	no	A string specifying the layout of the shipping label or "con-note" to be produced. Currently supported options are "A4" or "Thermal" (which is A6-size). The default is "A4".
goods	Array	yes	A list of goods to be picked up and delivered.
stops	Array	yes	A list of two stops: one each for pickup and delivery.

Params example:

```
{
  "customerCode": "DUMMY",
  "reference1": "ADMIN",
  "primaryService": "RF",
  "connoteFormat": "Thermal",
  "goods": [
    {
      "pieces": "1",
      "weight": "2",
      "width": "10",
      "height": "20",
      "depth": "30",
      "typeCode": "CTN"
    }
  ],
  "stops": [
    {
      "name": "Jack Watkins",
      "suburbName": "BRISBANE",
      "addressLine1": "10 William St",
      "addressLine2": "Building 2",
      "postCode": "4000",
      "state": "QLD",
      "instructions": "Ask receptionist to call contact",
      "contact": {
        "name": "Jane Wonder",
        "phone": "82937823",
        "email": "jane@email.com"
      },
      "timeWindow": {
        "earliest": "2025-01-01 10:00",

```

```

    "latest" : "2025-01-02 16:00"
  }
},
{
  "name" : "Name",
  "suburbName" : "Sydney",
  "addressLine1" : "15 Rose St",
  "postCode" : "2000",
  "state" : "NSW",
  "contact" : {
    "name" : "John Foster",
    "phone" : "94386738",
    "email" : "john.foster@email.com"
  }
}
]
}

```

Booking Result (BOOKING_RESULT)

Field name	Field type	Field description
jobNumber	String	Allocated job number.
jobDate	String	Allocated job date.
fee	Double	Price of the booking, excluding taxes
taxes	Array	List of taxes and surcharges.
etd	String	Estimated time-frame for delivery.
trackingNumber	String	Identification number which you can use to track this booking.
shippingLabel	String	Full contents of the shipping label file encoded as a Base64 string.

Result example:

```

{
  "jobNumber" : "20957",
  "jobDate" : "2015-01-01",
  "fee" : 12.97,
  "taxes" : [],
  "etd" : "n.a.",
  "trackingNumber" : "YY300205",
  "shippingLabel" : "...BASE64-encoded-file-bytes-here..."
}

```

3. Job Connote Scan (GET /booking/get-job-connote-scan)

This service is used to retrieve a scanned consignment note image associated with a job.

GET /booking/get-job-connote-scan		
Request	Parameters	CONNOTE_SCAN_PARAMS (query params)
	Headers	TEST: Authorization: Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	List of CONNOTE_SCAN_IMAGE objects

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/get-job-connote-scan>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/booking/get-job-connote-scan>

Connote Scan Params (CONNOTE_SCAN_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer
jobNumber	String	yes	Internal number that identifies the job.
jobDate	String	yes	The job's pick-up date using the format "yyyy-MM-dd"

Params example:

<https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/get-job-connote-scan?customerCode=DUMMY&jobNumber=32988&jobDate=2017-02-28>

Connote Scan Image (CONNOTE_SCAN_IMAGE)

Field name	Field type	Field description
------------	------------	-------------------

podImage	String	Scan image bytes encoded as Base64 string.
mime	String	MIME type of the scan image.

Result example:

```
[
  {
    "podImage": "...SUKqAAg...",
    "mime": "image/jpeg",
  },
  {
    "podImage": "...s22DDf45gsndgdfb...",
    "mime": "image/jpeg",
  }
]
```

4. Job POD (GET /booking/get-job-pod)

This service is used to retrieve a Proof of Delivery (POD) image associated with a job.

GET /booking/get-job-pod		
Request	Parameters	JOB_POD_PARAMS (query params)
	Headers	TEST: Authorization: Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	List of JOB_POD_IMAGE objects

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/get-job-pod>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/booking/get-job-pod>

Job POD Params (JOB_POD_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer
jobNumber	String	yes	Internal number that identifies the job.

jobDate	String	yes	The job's pick-up date using the format "yyyy-MM-dd"
---------	--------	-----	--

Params example:

```
https://sandbox.hunterexpress.com.au:443/sandbox/rest/hxws/booking/get-job-pod?customerCode=DU
MMY&jobNumber=32988&jobDate=2017-02-28
```

Job POD Image (JOB_POD_IMAGE)

Field name	Field type	Field description
podImage	String	POD image bytes encoded as Base64 string.
mime	String	MIME type of the POD image.

Result example:

```
[
  {
    "podImage": "...SUKqAAg...",
    "mime": "image/jpeg",
  },
  {
    "podImage": "...s22DDf45gsndgdfb...",
    "mime": "image/jpeg",
  }
]
```

5. Job Statuses (GET /booking/get-job-statuses)

This service is used to retrieve information about a job, as well as all available tracking data at the time of the call, provided as a list of status changes (e.g. Picked Up, Out for Delivery, Delivered, etc.)

GET /booking/get-job-statuses		
Request	Parameters	JOB_STATUSES_PARAMS (query params)

	Headers	<i>TEST:</i> Authorization:Basic aHh3czpoeHdz <i>PRODUCTION:</i> Authorization: Basic [your-key]
Response	Body	<u>JOB_STATUSES_RESPONSE</u>

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/booking/get-job-statuses>

PRODUCTION: <https://api.hunterexpress.com.au/LIVE/rest/hxws/booking/get-job-statuses>

Job Statuses Params (JOB_STATUSES_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer
trackingNumber	String	yes	A unique string that identifies a delivery job (e.g. YY296375)

Params example:

<https://sandbox.hunterexpress.com.au:443/sandbox/rest/hxws/booking/get-job-statuses?customerCode=DUMMY&trackingNumber=AA364999>

Job Statuses Response (JOB_STATUSES_RESPONSE)

Field name	Field type	Field description
number	Integer	Booking number
date	String	Pick-up date
reference1	String	Booking reference
costs	Array	List of <u>COST</u> objects
goods	Array	List of <u>GOODS</u> objects
stops	Array	List of <u>STOP</u> objects
hasPOD	Boolean	Indicates if this job contains proof of delivery
events	Arrays	List of <u>EVENT</u> objects

Result example:

```
{
  "number" : 20957,
  "date" : 1735664400000,
  "reference1" : "YY300205",
  "costs" : [
    {
      "serviceCode" : "RF",
      "serviceName" : "Road Freight",
      "quantity" : 2
    }
  ],
  "goods" : [
    {
      "pieces" : 1,
      "weight" : 2,
      "width" : 0,
      "height" : 0,
      "depth" : 0,
      "typeCode" : "ENV"
    }
  ],
  "stops" : [
    {
      "suburbName" : "BRISBANE,QLD",
      "postCode" : "4000",
      "state" : "QLD",
      "country" : "AU",
      "name" : "John Smith",
      "addressLine1" : "10 William St",
      "addressLine2" : "Building 2",
      "stopReference" : "AB-1234",
      "instructions" : "Ask receptionists to call the contact person",
      "contact" : "Jane Smith"
    },
    {
      "suburbName" : "SYDNEY",
      "postCode" : "2000",
      "state" : "NSW",
      "country" : "AU",
      "name" : "Jack Williams",
      "addressLine1" : "15 Rose St"
    }
  ],
  "hasPOD" : false,
  "events" : [
    {
      "when" : "2017-06-02 16:04:30",
      "what" : "On-Forwarded",
    }
  ]
}
```

```

    "who": "HXSyd",
    "info": ""
  },
  {
    "when": "2017-06-03 03:45:00",
    "what": "Scanned into Depot",
    "who": "TOWERS",
    "info": ""
  },
  {
    "when": "2017-06-03 13:12:00",
    "what": "Delivered",
    "who": "TOWERS",
    "info": "PoD: A KELLY"
  },
]
}

```

6. Services (GET /resources/services)

Returns the available services.

GET /resources/services		
Request	Parameters	SERVICES_PARAMS (query params)
	Headers	TEST: Authorization: Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	List of SERVICE objects

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/resources/get-services>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/resources/get-services>

Services Params (SERVICES_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	yes	A code that identifies the customer
serviceClass	String	no	The optional serviceClass argument is used to select services for nation-wide deliveries if set to 'FREIGHT', or

			omitted. If set to 'COURIER', the services apply to deliveries only in the metropolitan area implied by the account's address.
--	--	--	--

Params example:

```
https://sandbox.hunterexpress.com.au:443/sandbox/rest/hxws/resources/get-services?customerCode=DUMMY&serviceClass=FREIGHT
```

Result example:

```
[
  {
    "code" : "AF",
    "name" : "Air Freight",
    "primary" : true
  },
  {
    "code" : "RDL",
    "name" : "Re-Delivery",
    "primary" : true
  },
  {
    "code" : "RF",
    "name" : "Road Freight",
    "primary" : true
  },
  {
    "code" : "SDX",
    "name" : "Same Day Air Freight",
    "primary" : true
  }
]
```

7. Locations (GET /resources/get-locations)

Use this service to retrieve a list of valid suburb, postcode, state and country combinations for courier and freight services.

[Note: retrieving the full list of locations is relatively large and time consuming operation. Since the locations tend to change rarely there is absolutely no need to call this service every time you

need the locations. Instead, it's highly recommended to cache the locations and refresh it no more than once a day.]

GET /resources/get-locations		
Request	Parameters	LOCATIONS_PARAMS (query params)
	Headers	TEST: Authorization:Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	List of LOCATION objects

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/resources/get-locations>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/resources/get-locations>

Locations Params (LOCATIONS_PARAMS)

Field name	Field type	Mandatory	Field description
customerCode	String	no	A code that identifies the customer.
serviceClass	String	no	The optional serviceClass argument is used to select locations for nation-wide deliveries if set to ' FREIGHT ', or omitted. If set to ' COURIER ', the locations apply to deliveries only in the metropolitan area implied by the account's address.

Params example:

```
https://sandbox.hunterexpress.com.au:443/sandbox/rest/hxws/resources/get-locations?customerCode=DUMMY&serviceClass=FREIGHT
```

Result example:

```
[
  {
    "suburbName": "A1 MINE SETTLEMENT",
    "postCode": "3723",
    "state": "VIC",
    "country": "AU"
  },
  {
    "suburbName": "AARONS PASS",
```

```
"postCode": "2850",
"state": "NSW",
"country": "AU"
},
{
  "suburbName": "ABBA RIVER",
  "postCode": "6280",
  "state": "WA",
  "country": "AU"
}
]
```

8. Goods Types (GET /resources/get-goods-types)

Returns a list of valid goods types to be used when booking a job or getting a quote.

GET /resources/get-goods-types		
Request	Parameters	-
	Headers	TEST: Authorization: Basic aHh3czpoeHdz PRODUCTION: Authorization: Basic [your-key]
Response	Body	List of GOODS_TYPE objects

API endpoints:

TEST: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/resources/get-goods-types>

PRODUCTION: <https://api.hunterexpress.com.au/live/rest/hxws/resources/get-goods-types>

Query example:

```
https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/resources/get-goods-types
```

Result example:

```
[
  {
    "code" : "AUTOM",
    "description" : "Automotive/Car Part"
  },
  {
    "code" : "BULKS",
    "description" : "Bulky Item"
  },
  {
    "code" : "BULKL",
    "description" : "Bulky Item Very Large"
  }
]
```

Code Examples

The following code examples illustrate how to use the API:

Java: [java-code-samples.zip](#)

PHP: [php-code-samples.zip](#)

Note that you have to edit the example source code slightly before running it. Replace the "YOUR_AUTH_KEY" string with a valid authentication key which you received from Hunter Express.

Chrome Postman

You can also download a predefined collection of sample queries and import it into Chrome's Postman app:

<https://api.hunterexpress.com.au/sample-code/sandbox-samples.postman.json>

FAQ

What is the difference between the sandbox and live endpoints?

HX REST API has two different endpoints:

1) *Sandbox*: <https://sandbox.hunterexpress.com.au/sandbox/rest/hxws/>

All web services starting with this are for testing. You can use the sandbox endpoint to develop your REST client and safely make test bookings.

Everyone uses the same credentials to access the sandbox endpoint:

Username: hxws

Password: hxws

Customer/account code: DUMMY

2) *Live/production*: <https://api.hunterexpress.com.au/live/rest/hxws/>

Once you have finished your testing and development phase you will be given your live/production credentials. Simply change the sandbox endpoints in your REST client to the production ones and you are ready to go live.

Username: <supplied by HX>

Password: <supplied by HX>

Customer/account code: <your HX customer/account code>

APPENDIX A - ERROR CODES AND MESSAGES

If you receive a response with a **status code 400** the response body will contain a detailed message with one of the below codes/descriptions.

Error code	Description
ERR_MISSING_CONTACT_DATA	Each stop must have a contact.
ERR_MISSING_GOODS_DATA	Each booking must have goods.
ERR_MISSING_STOP_DATA	Each booking must have a stop.
WS_ERR_BADARG	Internal error: please contact HX support with full details.
WS_ERR_BADCRED	Customer's account is on stop credit. Please contact the HX accounts department.
WS_ERR_BADREF	The supplied reference is not valid.
WS_ERR_BADSUBURB	The supplied location is not valid.
WS_ERR_BADSVC	A supplied service code is not valid.
WS_ERR_BADTIME	Invalid pick-up time (outside of normal working hours)
WS_ERR_BADUSER	Supplied customer code is invalid
WS_ERR_NOIMAGE	The job doesn't have a POD image.
WS_ERR_NOJOB	Indicates that no job matching the specified criteria was found.
WS_ERR_NOPRICES	No services available for this location pair.
WS_ERR_NOREF	A required reference is missing.
WS_ERR_NOWEIGHT	Weight data is missing for a "FREIGHT" service
WS_ERR_ROCKYDB, WS_ERR_TNTDB	Internal error: please contact HX support with full details.
WS_ERR_SYSTEM	Internal error: please contact HX support with full details.
WS_ERR_TOOLATE	Specified pick-up time is too late for today.

APPENDIX B - COMMON REQUEST ERRORS

Here are a few examples of typical errors you may encounter. If you receive an HTTP status code other than 200, you'll find the details of the errors in the **response body**.

Bad Request (status code 400). You have missed the 'customerCode' query parameter.

```
{
  "webbitName": "",
  "errorCode": "400",
  "errorMessage": "Missing query parameter: [customerCode]",
  "errorOccurred": "2017-07-11 22:38:02"
}
```

Unauthorized (status code 401). Invalid or missing 'Authorization' header.

```
{
  "message": "Full authentication is required to access this resource",
  "status": "UNAUTHORIZED"
}
```

Not Found (status code 404). Incorrect or mistyped request endpoint.

```
{
  "webbitName": "",
  "errorCode": "404",
  "errorMessage": "Invalid request endpoint. Please check that you are using the correct URI to access this service.",
  "errorOccurred": "2017-07-11 15:32:30"
}
```

Unsupported Media Type (status code 415). You have used 'text/plain' instead of 'application/json' in a POST request which requires a body payload.

```
{
  "webbitName": "",
  "errorCode": "415",
  "errorMessage": "Unsupported media type [text/plain; charset=UTF-8]. Please referer to the API documentation. POST requests typically use 'application/json'",
  "errorOccurred": "2017-07-11 22:45:58"
}
```

Unsupported HTTP Method (status code 405). You have most likely used the http GET method instead of POST, or vice versa.

```
{
  "webbitName": "webcore-unsupported-http-method",
  "errorCode": "405-1",
  "errorMessage": "Unsupported HTTP method. Please refer to the API documentation and ensure that you are sending the request with the correct HTTP method (POST, GET).",
  "errorOccurred": "2017-12-07 14:33:56"
}
```