



Temando Api

Release 1.09

Developer Documentation

Temando IT Department

1st Edition July 2009

Contents

1. INTRODUCTION.....	3
1.1. PURPOSE	3
1.2. REVISION HISTORY.....	3
1.3. NAMESPACES	5
1.4. DOCUMENT CONVENTIONS	5
1.5. TERMINOLOGY	6
1.6. REFERENCES	7
2. API OVERVIEW	8
2.1. SOAP MESSAGES.....	9
2.2. USER AUTHENTICATION	13
2.3. ERROR HANDLING.....	14
2.4. RESTRICTIONS AND LIMITATIONS	15
2.4.1. Access Rights.....	15
2.4.2. General.....	15
2.5. PROCESS GUIDE	16
2.5.1. Get Quotes and Make a Booking (Client User).....	16
2.5.2. Retrieve and Book Requests (Carrier User).....	18
2.5.3. Retrieve/Register Client, Get Quotes and Make a Booking (Affiliate User).....	19
2.6. COMMON OPERATIONS.....	23
2.6.1. getQuotesByRequest.....	25
2.6.2. makeBookingByRequest	30
2.6.3. getRequest	38
2.6.4. getRequestsRequiringBooking.....	44
2.6.5. addBookingDetails	48
2.6.6. updateTrackingDetails	51
2.6.7. createClient	53
2.6.8. updateClient	56
2.6.9. getClient	58
2.6.10. getManifest	61
2.6.11. confirmManifest.....	68
2.6.12. cancelRequest.....	70
2.6.13. createLocation.....	72
2.6.14. updateLocation.....	74
2.6.15. getLocations	76
2.7. SOAP FAULTS.....	79
2.7.1. Err_Service01.....	79
2.7.2. Err_Service02.....	80
2.7.3. Err_Service03.....	80
2.7.4. Err_Connection01	80
2.7.5. Err_Connection02	81
2.7.6. Err_Connection03	81
2.7.7. Err_Access01.....	81
2.7.8. Err_Access02.....	82
2.7.9. Err_Access03.....	82
2.7.10. Err_Access04.....	83
2.7.11. Err_Access05.....	83
2.7.12. Err_Access06.....	83
2.7.13. Err_Access07.....	84
2.7.14. Err_Access08.....	84
2.7.15. Err_Access09.....	85
2.7.16. Err_Format	85
2.7.17. Err_Payment01	88

2.7.18.	<i>Err_Payment02</i>	88
2.7.19.	<i>Err_Payment03</i>	89
2.7.20.	<i>Err_Payment04</i>	89
2.7.21.	<i>Err_Permission01</i>	89
2.7.22.	<i>Err_Permission02</i>	90
2.7.23.	<i>Err_State01</i>	90
2.7.24.	<i>Err_State02</i>	91
2.7.25.	<i>Err_State03</i>	91
2.7.26.	<i>Err_State04</i>	91
2.7.27.	<i>Err_State05</i>	92
2.7.28.	<i>Err_State06</i>	92
2.7.29.	<i>Err_State07</i>	93
2.7.30.	<i>Err_State08</i>	93
2.7.31.	<i>Err_State09</i>	94
2.7.32.	<i>Err_State10</i>	94
2.7.33.	<i>Err_State11</i>	95
2.7.1.	<i>Err_State12</i>	95
2.7.2.	<i>Err_Data01</i>	96
2.8.	COMMON TYPES	98
2.8.1.	<i>Compound Data Types</i>	98
2.8.2.	<i>Simple Data Types</i>	146
3.	APPENDIX A: SOAPUI SUPPORT	177
3.1.	SETUP	177
3.2.	EXAMPLE CALL	179
4.	APPENDIX B: PHP EXAMPLE	1
4.1.	OVERVIEW	1
4.2.	SOAP INSTALLATION TESTING	1
4.3.	CODE	1
4.4.	SCREENSHOTS	12
4.4.1.	<i>Success</i>	12
4.4.2.	<i>Error (exception)</i>	14
5.	APPENDIX C: .NET EXAMPLE	15
5.1.	OVERVIEW	15
5.2.	CODE	15
5.3.	SETUP	15
5.4.	KNOWN .NET ISSUES	17
5.4.1.	<i>Adapter Defect – Cannot Resolve/Import</i>	17
5.4.2.	<i>Adapter Issue – Authentication</i>	18
6.	APPENDIX D: INSURANCE GUIDELINES	19
6.1.	GUIDELINES	19
6.1.1.	<i>Client</i>	19
6.1.2.	<i>Affiliate</i>	22

1. Introduction

1.1. Purpose

This document constitutes the normative definition of the Temando API Release 1. This version supports quoting and booking freight as well as various general information operations (eg. vehicle types, etc).

1.2. Revision History

Version	Date	Changes
Release 1	July 2009	Release of the API documentation.
Release 1.01	17 th September 2009	<p>Delivery method is no longer from a predefined set of options but is defined by the Carrier.</p> <p>Added support Appendices for using soapUI and example PHP code.</p> <p>Modified and added tags for Location data type.</p> <p>New Soap fault added for invalid date/locations (eg. can not pickup because of public holidays) - Err_Data01.</p> <p>Addition of consignment document for calls.</p> <p>Added a new operation <code>getRequestsRequiringBooking</code>.</p> <p>Addition of postal box and unattended questions for 2.8.1.3 Anywhere details.</p> <p>Addition of Carrier Id for 2.8.1.8 Carrier details and its replacement of the <code>companyName</code> field in the 2.8.1.14 BookingQuote.</p>
Release 1.02	9 th November 2009	Addition of tracking details in the 2.8.1.6 AvailableQuote details.
Release 1.03	17 th February 2010	Addition of instructions (2.8.2.33 Instructions) in the 2.8.1.1 Request details.
Release 1.04	5 th May 2010	<p>Addition of <code>createClient</code> (2.6.7 <code>createClient</code>), <code>updateClient</code> (2.6.8 <code>updateClient</code>) and <code>getClient</code> (2.6.9 <code>getClient</code>) operations.</p> <p>Modification of <code>getQuotesByRequest</code> (2.6.1 <code>getQuotesByRequest</code>), <code>makeBookingByRequest</code> (2.6.2 <code>makeBookingByRequest</code>) and <code>getRequest</code> (2.6.3 <code>getRequest</code>) operations to allow all users with</p>

		API (Affiliate): General user rights access to call these operations.
Release 1.05	25 th May 2010	<p>Addition of updateTrackingDetails (2.6.6 updateTrackingDetails), lodgeDispatch (2.6.10 getManifest) and cancelRequest (2.6.12 cancelRequest) operations.</p> <p>Removed getRequestByIdByReference operation. Removal of the access rights API (Client): Plugin and API (Carrier): Plugin.</p> <p>Addition of .NET example.</p>
Release 1.06	23 rd June 2010	<p>The disabling of the elements: destinationBusNotifyBefore, destinationBusTailgateLifter, destinationResNotifyBefore, destinationResTailgateLifter, originBusNotifyBefore, originBusTailgateLifter, originResNotifyBefore and originResTailgateLifter. These elements are located within the Anywhere type (2.8.1.3 Anywhere).</p>
Release 1.07	31 st August 2010	<p>Addition of General element (2.8.1.5 General). Contains goodsValue element used for calculating insurance premiums.</p> <p>Addition of extras. These elements are located within the AvailableQuote and BookingQuote type (2.8.1.6 AvailableQuote and 2.8.1.14 BookingQuote).</p>
Release 1.08	31 st March 2011	<p>Addition of createLocation (2.6.13 createLocation), updateLocation (2.6.14 updateLocation) and getLocation (2.6.15 getLocations) operations.</p> <p>Addition of getManifest () operation.</p> <p>Removal of lodgeRequest and makeBookingById operations.</p> <p>Addition of reference (2.8.2.102 ClientReference) in the 2.8.1.1 Request details.</p> <p>Addition of detail element for getRequest operation (2.6.3 getRequest). Allows response to be summarised or detailed.</p>
Release 1.09	24 th June 2011	<p>Addition of label document for getRequest (2.6.3 getRequest), makeBookingByRequest (2.6.2 makeBookingByRequest) and getManifest (2.6.10 getManifest) operations.</p> <p>Addition of confirmManifest () operation.</p>

1.3. Namespaces

The following xml namespaces and prefixes are used throughout this document:

Prefix	Namespace	Source
soap	http://schemas.xmlsoap.org/wsdl/soap/	SOAP
soapenc	http://schemas.xmlsoap.org/soap/encoding	SOAP
wsdl	http://schemas.xmlsoap.org/wsdl/	WSDL
wsse	http://schemas.xmlsoap.org/ws/2002/04/secext	WS-Security
xsi	http://www.w3.org/2001/XMLSchema-instance	XMLSchema
xsd	http://www.w3.org/2001/XMLSchema	XMLSchema
tem	http://api.temando.com/schema/2009_06/server.xsd	Temando API
com	http://api.temando.com/schema/2009_06/common.xsd	Temando API

1.4. Document Conventions

Example code is written in a fixed-width sans font, ie “example”, and does not include the surrounding quotes.

XML schema extracts are contained in a green box in a fixed-width font, and do not include namespace declarations (all prefixes are defined above in section 1.3 Namespaces). For example:

```
<xsd:complexType name="Inclusion">
  <xsd:all>
    <xsd:element name="summary">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="50"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="details">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:maxLength value="500"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
```

Block XML examples are contained in a blue box, and similarly do not include namespace declarations. Examples of complex types will include an xsi:type attribute on the top-level element for clarity, but this is neither required on request messages nor generated on response messages. Eg:

```
<inclusion xsi:type="com:Inclusion">
  <summary>Tracking available</summary>
```

```
<details>Tracking information will be available for this quote.</details>
</inclusion>
```

1.5. Terminology

The following table defines the terminology and acronyms used in this document:

Term/Acronym	Definition
Quote	An individual quote for a carrier to provide the service.
Request	The details of the item/s to be sent.
Booking	A quote that has been accepted and payment received.
Booking Number	Booking number allocated by the corresponding Carrier for the booking.
Carrier	A transportation company that provides the quote for the transportation of freight.
GST	Goods and Services Tax. 10% tax levied on all Australian freight (but not international). Note that the full tax situation is sufficiently complicated to be beyond the scope of this document.
HTTP	Hyper Text Transfer Protocol
Local time	As used in this document, "Local Time" always refers to the local time specified against the users Temando account.
SOAP	Simple Object Access Protocol – A Specification for XML-based communication ¹ .
WSDL	Web Services Description Language – A specification for describing web services in terms of messages and operations. A copy of the WSDL schema for the Temando API is available from http://api.temando.com/schema/2009_06/server.wsdl
XML	eXtensible Markup Language – A method of formatting structured data in a way that purports to be human-readable.
XML Schema	A data / structure model for XML allowing specification of fully typed complex data structures. The schemas defining the Temando API are available at http://api.temando.com/schema/2009_06/common.xsd

1.6. References

- [AXML] Bray T “*The Annotated XML 1.0 Specification*” Sep 1998
<http://www.xml.com/axml/testaxml.htm>
- [HP-Roman-8] WRQ “Character Sets for HP Emulation” Tech note 1179
May 1998. <http://support.wrq.com/techdocs/1179.html>
- [HTTP] Fielding R, Gettys J, Mogul J, Frystyk H et al “Hypertext Transfer Protocol – HTTP/1.1” IETF RFC2616, Jun 1999.
<http://www.ietf.org/rfc/rfc2616.txt>
- [SOAP] Box D, Ehnebuske D, Kakivaya G, Layman A et al “*Simple Object Access Protocol (SOAP) 1.1*” W3C Note May 2000.
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [WS-Security] Atkinson B, Della-Libera G, Hada S, Hondo M et al “*Web Services Security (WS-Security) 1.0*” IBM Apr 2002. <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>
- [WSDL] Christensen E, Curbera F, Meredith G and Weerawarana S “*Web Services Description Language (WSDL) 1.1*” W3C Note, Mar 2001.
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [XML] Bray T, Paoli J, Sperberg-McQueen C M, Maler E et al “*Extensible Markup Language (XML) 1.0 (Third Edition)*” W3C Recommendation Feb 2004. <http://www.w3.org/TR/2004/REC-xml-20040204>
- [XML1.1] Bray T, Paoli J, Sperberg-McQueen C M, Maler E et al “*Extensible Markup Language (XML) 1.1*” W3C Recommendation Feb 2004. <http://www.w3.org/TR/2004/REC-xml11-20040204>
- [XMLNS] Bray T, Hollander D, Layman A, Tobin R “*Namespaces in XML 1.1*” W3C Recommendation Feb 2004. <http://www.w3.org/TR/2004/REC-xml-names11-20040204>
- [XMLSchema-1] Thompson H S, Beech D, Maloney M, Mendelsohn N “*XML Schema Part 1: Structures (Second Edition)*” W3C Recommendation Oct 2004. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- [XMLSchema-2] Biron P V, Malhotra A “*XML Schema Part 2: Datatypes (Second Edition)*” W3C Recommendation Oct 2004.
<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>
- [XMLSIG] Eastlake D, Reagle J, Solo D, Bartel M et al “*XML-Signature Syntax and Processing*” W3C Recommendation Feb 2002.
<http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>

2. API Overview

Communication with the API uses a simple request/response protocol based on SOAP requests over a HTTP transport layer. The request message is submitted to the API server with a POST request to the API URI – the server responds with the response message as the content body.

2.1. SOAP Messages

Anatomy of a soap request message (getRequest in this instance)²:

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <soapenv:Envelope
3.     xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
4.     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6.   <soapenv:Header>
7.     <wsse:Security
8.         xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
9.       <wsse:UsernameToken>
10.        <wsse:Username>username</wsse:Username>
11.        <wsse:Password>password</wsse:Password>
12.      </wsse:UsernameToken>
13.    </wsse:Security>
14.  </soapenv:Header>
15.  <soapenv:Body>
16.    <tem:getRequest
17.        xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
18.      <requestId>100000</requestId>
19.      <detail>Detailed</detail>
20.    </tem:getRequest>
21.  </soapenv:Body>
22. </soapenv:Envelope>

```

Line 1: XML header declaration (recommended but technically optional³. In the absence of a declaration the message is assumed to use XML 1.0 (either 1.0 or 1.1 is supported). If the header is omitted, or is present but specifies no character encoding, the message MUST be encoded in UTF-8. Note that irrespective of the message's encoding, the contents of the message must be mappable to the HP Roman-8 character set (see below in "Restrictions and Limitations").

Currently supported character sets are UTF-8, UTF-16 (big or little endian), ISO-8859-1 and US-ASCII.

Lines 2-5,22: SOAP envelope wrapper – top-level document element.

Lines 6-14: SOAP headers. Currently the only recognized header is the WS-Security Security header, which must be specified in all requests.

Lines 15,21: SOAP body. Contains the actual request message from one of the operations defined in this document.

Lines 16-20: Main request body (in this case a getRequest request).

The response body is similar but does not currently include any SOAP headers.

Example SOAP response to getRequestResponse:

```
<?xml version="1.0" encoding="UTF-8"?>
```

² Actual SOAP messages not colour coded.

³ "XML documents SHOULD begin with an XML declaration which specifies the version of XML being used." [XML].

```

<soap-env:Envelope xmlns:soap-
env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getRequestResponse id="100000"
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
      <status>Completed</status>
      <anythings>
        <anything>
          <class>Vehicle</class>
          <subclass>Car</subclass>
          <qualifierVehicleMake>Ford</qualifierVehicleMake>
          <qualifierVehicleModel>Mustang</qualifierVehicleModel>
          <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
            289</qualifierVehicleDescription>
          <qualifierVehicleRunning>Y</qualifierVehicleRunning>
          <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
          <qualifierVehicleYear>1965</qualifierVehicleYear>
          <quantity>1</quantity>
        </anything>
      </anythings>
      <anywhere>
        <itemNature>Domestic</itemNature>
        <itemMethod>Door to Door</itemMethod>
        <originCountry>AU</originCountry>
        <originCode>4000</originCode>
        <originSuburb>Brisbane</originSuburb>
        <destinationCountry>AU</destinationCountry>
        <destinationCode>2000</destinationCode>
        <destinationSuburb>Sydney</destinationSuburb>
        <destinationIs>Residence</destinationIs>
        <destinationResPostalBox>N</destinationResPostalBox>
        <destinationResUnattended>N</destinationResUnattended>
        <destinationResInside>N</destinationResInside>
        <destinationResLimitedAccess>N</destinationResLimitedAccess>
        <destinationResHeavyLift>N</destinationResHeavyLift>
        <originIs>Business</originIs>
        <originBusUnattended>N</originBusUnattended>
        <originBusDock>Y</originBusDock>
        <originBusForklift>N</originBusForklift>
        <originBusLoadingFacilities>N</originBusLoadingFacilities>
        <originBusInside>N</originBusInside>
        <originBusLimitedAccess>N</originBusLimitedAccess>
        <originBusHeavyLift>N</originBusHeavyLift>
        <originBusTailgateLifter>N</originBusTailgateLifter>
        <originBusContainersSwingLifter>N</originBusContainersSwingLifter>
      </anywhere>
      <anytime>
        <readyDate>2009-07-01</readyDate>
        <readyTime>PM</readyTime>
      </anytime>
      <general>
        <goodsValue>2000.00</goodsValue>
      </general>
      <origin>
        <description>ABC Suppliers</description>
        <contactName>John Doe</contactName>
        <companyName>ABC Suppliers</companyName>
        <street>1 Originating Street</street>
        <suburb>Brisbane</suburb>
        <state>QLD</state>
        <code>4000</code>
        <country>AU</country>
        <phone1>(07) 3333 3333</phone1>
        <phone2></phone2>
        <fax>(07) 3333 3331</fax>
        <email>bneoffice@abcsuppliers.com.au</email>
      </origin>
    </ns1:getRequestResponse>
  </soap-env:Body>
</soap-env:Envelope>

```

```

</origin>
<destination>
  <contactName>Jane Doe</contactName>
  <street>2 Recipient Street</street>
  <suburb>Sydney</suburb>
  <state>NSW</state>
  <code>2000</code>
  <country>AU</country>
  <phone1>(02) 9333 3333</phone1>
  <phone2></phone2>
  <fax>(02) 9333 3331</fax>
  <email>jane.doe@yahoo.com.au</email>
</destination>
<quotes>
  <quote quote id="111111">
    <generated>Automatically</generated>
    <accepted>N</accepted>
    <bookingNumber>XYZ300001</bookingNumber>
    <consignmentNumber>UVW400001</consignmentNumber>
    <consignmentDocument>JVBER ... shortened for readability
      ... XRhbG</consignmentDocument>
    <consignmentDocumentType>application/pdf</consignmentDocumentType>
    <labelDocument>e3Rf3 ... shortened for readability ... X8ggy
      </labelDocument>
    <labelDocumentType>application/pdf</labelDocumentType>
    <trackingStatus>In Transit</trackingStatus>
    <trackingStatusOccurred>2009-07-02T15:25:46Z
      </trackingStatusOccurred>
    <trackingLastChecked>2009-07-02T18:13:12Z</trackingLastChecked>
    <trackingFurtherDetails></trackingFurtherDetails>
    <totalPrice>110.00</totalPrice>
    <basePrice>100.00</basePrice>
    <tax>10.00</tax>
    <currency>AUD</currency>
    <deliveryMethod>Off-peak</deliveryMethod>
    <usingGeneralRail>N</usingGeneralRail>
    <usingGeneralRoad>Y</usingGeneralRoad>
    <usingGeneralSea>N</usingGeneralSea>
    <usingExpressAir>N</usingExpressAir>
    <usingExpressRoad>N</usingExpressRoad>
    <etaFrom>2</etaFrom>
    <etaTo>3</etaTo>
    <guaranteedEta>Y</guaranteedEta>
  </quote>
  <adjustments>
    <adjustment>
      <description>Credit Card Payment Adjustment</description>
      <amount>2.00</amount>
      <tax>0.20</tax>
    </adjustment>
    <adjustment>
      <description>Credit Payment Adjustment</description>
      <amount>-2.00</amount>
      <tax>-0.20</tax>
    </adjustment>
  </adjustments>
  <inclusions>
    <inclusion>
      <summary>Tracking available</summary>
      <details>Tracking information will be available for this
        quote.</details>
    </inclusion>
    <inclusion>
      <summary>Automatic Booking</summary>
      <details>Automatic booking generated.</details>
    </inclusion>
    <inclusion>
      <summary>Automatic Consignment</summary>
      <details>Automatic consignment note generated.</details>
    </inclusion>
  </inclusions>
</quotes>

```

```

</inclusions>
<extras>
  <extra>
    <summary>Insurance</summary>
    <details>Insurance cover up to $5,000.</details>
    <totalPrice>22.00</totalPrice>
    <basePrice>20.00</basePrice>
    <tax>2.00</tax>
    <mandatory>Y</mandatory>
    <adjustments>
      <adjustment>
        <description>Credit Card Payment Adjustment</description>
        <amount>0.40</amount>
        <tax>0.04</tax>
      </adjustment>
    </adjustments>
  </extra>
</extras>
<carrier>
  <id>654321</id>
  <companyName>ZYX Transport</companyName>
  <companyContact>John Doe</companyContact>
  <streetAddress>1 Transport Street</streetAddress>
  <streetSuburb>Brisbane</streetSuburb>
  <streetCity>Brisbane</streetCity>
  <streetState>QLD</streetState>
  <streetCode>4000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 1111</postalAddress>
  <postalSuburb>Brisbane</postalSuburb>
  <postalCity>Brisbane</postalCity>
  <postalState>QLD</postalState>
  <postalCode>4001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(07) 3333 3333</phone1>
  <email>john.doe@zyxtransport.com.au</email>
  <website>https://www.zyxtransport.com.au</website>
  <conditions>Our conditions include: lots of text.</conditions>
</carrier>
</quote>
</quotes>
<instructions>Please use side door.</instructions>
<reference>ABC200000</reference>
<comments>Please don't use any chains on the axle.</comments>
</ns1:getRequestResponse>
</soap-env:Body>
</soap-env:Envelope>

```

2.2. User Authentication

User authentication is performed using the WS-Security standard SOAP headers [WS-Security], which must be included in every request message. Presently only username/password authentication is supported, however support for SSL certificate based authentication and signed xml [XMLSIG] are intended for a future release. Users may further be restricted by IP address.

Example WS-Security header block (occurs within the SOAP header):

```
<wsse:Security
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
  <wsse:UsernameToken>
    <wsse:Username>username</wsse:Username>
    <wsse:Password>password</wsse:Password>
  </wsse:UsernameToken>
</wsse:Security>
```

An accepted security header consists of a single Security element containing exactly one UsernameToken, which in turn contains Username and Password elements. The Username and Password elements in turn supply the user credentials in plain text (for this reason, an SSL connection is required for transport-level security).

See [WS-Security] for more information (but note that the portions of the specification not documented here are not currently implemented).

If authentication fails for whatever reason, the server will return a SOAP Fault indicating the failure as defined within section 2.7 SOAP Faults (specifically [Err Access](#) errors).

Example authentication fault:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access03</faultcode>
      <faultstring>The login details you have provided are not a valid
        Username and Password combination.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.3. Error Handling

Errors caused by an invalid request message will generate a SOAP fault in the response body and the operation will have no further effect (see 2.7 SOAP Faults for a list of standard fault codes).

2.4. Restrictions and Limitations

All API users must have at least one registered access right (see below) to gain access to any API operations.

2.4.1. Access Rights

The API's operations are restricted based on a user's API access rights.

The following are the available API access rights:

- API (Client): General
- API (Carrier): General
- API (Affiliate): General

2.4.2. General

There are currently no restrictions or limitations on the service requests themselves or returned data (eg. maximum number of quotes, etc.) but this may change in the future.

2.5. Process Guide

This section provides a step-by-step guide to some of the most common processes that would be implemented by users of the API.

Note that these are high level guides to the overall processes and do not contain specific details about the actual elements and structures to be sent in a request or received in a response. This information is available in the 2.6 Common Operations.

2.5.1. Get Quotes and Make a Booking (Client User)

The following process would be used when a user wants to check what quotes are available based on particular request parameters and then book in a request using one of the available quotes.

Note: examples of the actual elements and their values are available within the operation descriptions (references provided in the steps below).

2.5.1.1. Step 1. Get Quotes

The first step is to identify what quotes are available. To achieve this a call must be made to the `getQuotesByRequest` (see section 2.6.1 `getQuotesByRequest`) operation and provide the particulars of the request that needs to be sent. These particulars are grouped as follows:

Anythings	The details about the item/s being sent.
Anywhere	The details about where the item/s are being sent from/to.
Anytime	The details about when the item/s are being sent.
General	The general details about the item/s are being sent (eg. goods value).

Once this operation has been processed the response will contain a list of available quotes. For each quote returned the following information is provided:

Quote	General information for the quote, such as price, delivery time, etc.
Adjustments	List of adjustments that will be applied based on the payment method used at the booking stage.

Inclusions	If there are any inclusions associated with the quote (eg. tracking available, instant booking, etc). These incur no extra expense as they are included in the quote price.
Extras	If there are any extras available for the quote (eg. insurance, etc). These are an extra expense and will be added to the quote price if selected in the booking step.
Carrier	Information about the carrier service.

2.5.1.2. Step 2. Make a Booking

Once the available quotes have been provided in Step 1 and the required quote identified a booking must be made to register the request and the identified quote. To achieve this a call must be made to the `makeBookingByRequest` (see section 2.6.2 `makeBookingByRequest`) operation. The information required for this call is grouped as follows:

Anythings	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Anywhere	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Anytime	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
General	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Origin	The specific details about where an item/s is being sent from or the originating address.
Destination	The specific details about where an item/s is being sent to or the final destination address.
Quote	<p>This will include:</p> <ul style="list-style-type: none">• The specific details about the accepted quote.• The extras to be included with the quote (eg. insurance, etc). <p>These must match exactly with the details provided in the response to the <code>getQuotesByRequest</code> call in step 1.</p>
Payment	The payment details for the payment of the quote.
Instructions	Delivery instructions that should be communicated to the driver.
Comments	General comments relating to the request.
PromotionCode	Temando code used for promotional partners.

Once this operation has been processed the response will contain the following information:

Request Id	The unique Temando id for this booking.
Booking Number	The carrier's booking number for the request. This is only provided if it was detailed as an inclusion for the quote that was accepted from step 1.
Consignment Details	This will include the consignment number, consignment document and document type. This is only provided if it was detailed as an inclusion for the quote that was accepted from step 1.

2.5.2. Retrieve and Book Requests (Carrier User)

The following process would be used when a carrier wants to retrieve those requests that are awaiting booking and then to register a booking number (and other details if available) for these request.

Note: examples of the actual elements and their values are available within the operation descriptions (references provided in the steps below).

2.5.2.1. Step 1. Retrieve Requests

The first step is to identify what requests require booking. To achieve this a call must be made to the `getRequestsRequiringBooking` (see section 2.6.4 `getRequestsRequiringBooking`) operation. No information needs to be provided to this call as your carrier account will be identified from your login information.

Once this operation has been processed the response will contain a list of requests requiring booking. For each request returned the following information is provided:

Anythings	The details about the item/s being sent.
Anywhere	The details about where the item/s are being sent from/to. Only the details that were required for rate calculations (eg. postal code, suburb, etc).
Anytime	The details about when the item/s are being sent.
General	The general details about the item/s are being sent (eg. goods value).
Origin	The specific details about where an item/s is being sent from or the originating address.

Destination	The specific details about where an item/s is being sent to or the final destination address.
Quote	<p>This will include:</p> <ul style="list-style-type: none">• General information for the quote, such as price, delivery time, etc.• If there are any inclusion associated with the quote (eg. tracking available, instant booking, etc).• If there are any extras associated with the quote (eg. insurance, etc).• Information about your carrier service that was provided to the client.
Instructions	Delivery instructions that should be communicated to the driver.
Comments	General comments relating to the request.

2.5.2.2. Step 2. Book

Once the requests that are awaiting booking have been provided in Step 1 for each of these requests you will need to register your booking details. To achieve this a call must be made to the addBookingDetails (see section 2.6.5 addBookingDetails) operation for each individual request. The information required for this call is grouped as follows:

Booking Number	Your carrier service's unique booking number. This is a mandatory entry.
Consignment Details	This can include the consignment number, consignment document and document type. These are not mandatory entries.

Once this operation has been processed no response will be returned.

2.5.3. Retrieve/Register Client, Get Quotes and Make a Booking (Affiliate User)

The following process would be used when an affiliate wants to retrieve or register a client's details, then using this client's account check what quotes are available based on particular request parameters and then book in a request using one of the available quotes.

Note: examples of the actual elements and their values are available within the operation descriptions (references provided in the steps below).

2.5.3.1. Step 1. Retrieve/Register Client

The first step is to identify if a client already has an account with Temando. If the client does have an account then you will retrieve their account details, however if not you will need to register their details with Temando.

Already Has an Account

If a client already has an account with Temando they will have a login id and password. This information can be sent with the getClient (see section 2.6.9 getClient) operation to retrieve their details, in particular their client id which is required for the further steps.

Once this operation has been processed the response will contain the client's details.

Client Details	General information about the client, such as id, name, address, etc.
-----------------------	---

Does Not Have an Account

If a client does not already have an account with Temando they will need to have an account created. To achieve this a call must be made to the createClient (see section 2.6.7 createClient) operation and provide the particulars of the new client. These particulars are grouped as follows:

Login Details	The client's supplied login id and password.
Client Details	General information about the client, such as name, address, phone, email address, etc.
PromotionCode	Temando code used for promotional partners.

Once this operation has been processed the response will contain a client id.

Client Id	The unique Temando id for a client.
------------------	-------------------------------------

2.5.3.2. Step 2. Get Quotes

The next step is to identify what quotes are available. To achieve this a call must be made to the getQuotesByRequest (see section 2.6.1 getQuotesByRequest) operation and provide the particulars of the request that needs to be sent along with the client id extracted in step 1. These particulars are grouped as follows:

Anythings	The details about the item/s being sent.
Anywhere	The details about where the item/s are being sent from/to.
Anytime	The details about when the item/s are being sent.

General	The general details about the item/s are being sent (eg. goods value).
Client Id	The unique Temando id for a client.

Once this operation has been processed the response will contain a list of available quotes. For each quote returned the following information is provided:

Quote	General information for the quote, such as price, delivery time, etc.
Adjustments	List of adjustments that will be applied based on the payment method used at the booking stage.
Inclusions	If there are any inclusions associated with the quote (eg. tracking available, instant booking, etc). These incur no extra expense as they are included in the quote price.
Extras	If there are any extras available for the quote (eg. insurance, etc). These are an extra expense and will be added to the quote price if selected in the booking step.
Carrier	Information about the carrier service.

2.5.3.3. Step 3. Make a Booking

Once the available quotes have been provided in Step 1 and the required quote identified a booking must be made to register the request and the identified quote. To achieve this a call must be made to the `makeBookingByRequest` (see section 2.6.2 `makeBookingByRequest`) operation. The information required for this call is grouped as follows:

Anythings	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Anywhere	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Anytime	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
General	This is the exact same details as what was provided in the <code>getQuotesByRequest</code> call in step 1.
Origin	The specific details about where an item/s is being sent from or the originating address.
Destination	The specific details about where an item/s is being sent to or the final destination address.
Quote	This will include: <ul style="list-style-type: none">• The specific details about the accepted quote.

- The extras to be included with the quote (eg. insurance, etc).

These must match exactly with the details provided in the response to the getQuotesByRequest call in step 1.

Payment	The payment details for the payment of the quote.
Instructions	Delivery instructions that should be communicated to the driver.
Comments	General comments relating to the request.
PromotionCode	Temando code used for promotions or identifying sales representatives.
Client Id	The unique Temando id for a client.

Once this operation has been processed the response will contain the following information:

Request Id	The unique Temando id for this booking.
Booking Number	The carrier's booking number for the request. This is only provided if it was detailed as an inclusion for the quote that was accepted from step 1.
Consignment Details	This will include the consignment number, consignment document and document type. This is only provided if it was detailed as an inclusion for the quote that was accepted from step 1.

2.6. Common Operations

This section describes the API operations used in the basic quoting and booking process, or associated with it. The normative schema definitions of these elements are contained in the http://api.temando.com/schema/2009_06/server.xsd schema.

getQuotesByRequest:	Identify available quotes for the specific request details.
makeBookingByRequest:	Create a new request, accept (ie. book) a quote and process the supplied payment details.
getRequest:	Retrieve the details of a request (including associated quotes) stored in the Temando system.
getRequestRequiringBooking:	Retrieve a list of all request that requiring booking by a carrier.
addBookingDetails:	Adds a carrier's booking details to a request.
updateTrackingDetails:	Updates a carrier's tracking details for a request.
createClient:	This operation creates a new client account.
updateClient:	Updates a client's account details.
getClient:	Retrieve the details of a client's account details stored in the Temando system.
lodgeDispatch:	Adds a carrier's dispatch details to a request.
cancelRequest:	Cancels a request.

The typical quoting and booking process looks something like this:

1. User identifies all the information required to identify available quotes. Includes the request details: anything/s (item/s details), anywhere (basic to and from) and anytime (when).
2. System locates available quotes matching the user's request (**getQuotesByRequest**).
3. User selects the desired quote

4. User supplies the selected quote; the full details of the request: anything/s (item/s details), anywhere (basic to and from), anytime (when), origin (detailed from), destination (detailed to), delivery instructions and comments; payment details (**makeBookingByRequest**). System books the quote and returns the request id to the user (if successful).

4a. If the booking fails (payment declined, quote no longer available, etc), user will be required to either try again (ie. step 4. if payment issue or step 2. if quote no longer available).

2.6.1. getQuotesByRequest

The `getQuotesByRequest` operation retrieves all available quotes (automatically generated from a carrier's rate information) based on the supplied request details.

The response message is a `getQuotesByRequestResponse`, and contains a list of all available quotes. Quote results are not listed in any particular order.

If a query contains invalid data then a `SoapFault` is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.1.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.1.2. Request – getQuotesByRequest

```
<xsd:element name="getQuotesByRequest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="anythings">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="anything" type="com:Anything" maxOccurs="10"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="anywhere" type="com:Anywhere"/>
      <xsd:element name="anytime" type="com:Anytime" minOccurs="0"/>
      <xsd:element name="general" type="com:General" minOccurs="0"/>
      <xsd:element name="origin" type="com:Location"/>
      <xsd:element name="destination" type="com:Location"/>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child content:

The full details of a request. This includes all the details about what is being sent, where it is being sent from and to, and when it is to be sent.

Field	Base type	Occurs	Description
anythings	list	Always	The details about the item/s being sent. See section 2.8.1.2 Anything for further details.
anywhere	complex	Always	The details about where the item/s are being sent from/to. See section 2.8.1.3 Anywhere for further details.
anytime	complex	Optional	The details about when the

			item/s are being sent. If not specified defaults to the next available date (ie. ASAP). See section 2.8.1.4 Anytime for further details.
general	complex	Optional	The general details about the item/s being sent. See section 2.8.1.5 General for further details.
origin	complex	Optional	The details about where an item/s is being sent from or the originating address. Any matching values from the anywhere element (if supplied) will override those values supplied in this element. See section 2.8.1.12 Location for further details.
destination	complex	Optional	The details about where an item/s is being sent to or the final destination address. Any matching values from the anywhere element (if supplied) will override those values supplied in this element. See section 2.8.1.12 Location for further details.
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
clientId	<ul style="list-style-type: none"> Only applies for users with <i>API (Affiliate): General</i> access rights. Any supplied values are disregarded in all other cases. Mandatory for users with <i>API (Affiliate): General</i> access rights.

Example:

Request quotes for sending a Ford, Mustang, GT Convertible 2dr Auto 3sp 289, 1965 (running but not registered), door-to-door, from a business in Brisbane (4000) (with notifications before pick-up) to a business in Sydney (2000) with pickup on 01 July 2009 in the morning.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
      <tem:getQuotesByRequest
        xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
        <anythings>
          <anything>
            <class>Vehicle</class>
            <subclass>Car</subclass>
            <qualifierVehicleMake>Ford</qualifierVehicleMake>
            <qualifierVehicleModel>Mustang</qualifierVehicleModel>
            <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
              289</qualifierVehicleDescription>
            <qualifierVehicleRunning>Y</qualifierVehicleRunning>
            <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
            <qualifierVehicleYear>1965</qualifierVehicleYear>
            <quantity>1</quantity>
          </anything>
        </anythings>
        <anywhere>
          <itemNature>Domestic</itemNature>
          <itemMethod>Door to Door</itemMethod>
          <originCountry>AU</originCountry>
          <originCode>4000</originCode>
          <originSuburb>Brisbane</originSuburb>
          <destinationCountry>AU</destinationCountry>
          <destinationCode>2000</destinationCode>
          <destinationSuburb>Sydney</destinationSuburb>
          <destinationIs>Residence</destinationIs>
          <destinationResNotifyBefore>N</destinationResNotifyBefore>
          <destinationResLimitedAccess>N</destinationResLimitedAccess>
          <originIs>Business</originIs>
          <originBusNotifyBefore>Y</originBusNotifyBefore>
          <originBusLimitedAccess>N</originBusLimitedAccess>
        </anywhere>
        <anytime>
          <readyDate>2009-07-01</readyDate>
          <readyTime>PM</readyTime>
        </anytime>
        <general>
          <goodsValue>2000.00</goodsValue>
        </general>
      </tem:getQuotesByRequest>
    </soapenv:Body>
  </soapenv:Envelope>
```

2.6.1.3. Response - getQuotesByRequestResponse

```
<xsd:element name="getQuotesByRequestResponse">
```

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="quote" type="com:AvailableQuote"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Child Content:

Field	Type	Occurs	Description
quote	list	Any	A list of available quotes matching a particular request. See section 2.8.1.6 AvailableQuote for further details.

Example:

The quote available for a door-to-door request.

```
<soap-env:Envelope
  xmlns:soap-env=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getQuotesByRequestResponse
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
      <quote>
        <generated>Automatically</generated>
        <accepted>N</accepted>
        <totalPrice>110.00</totalPrice>
        <basePrice>100.00</basePrice>
        <tax>10.00</tax>
        <currency>AUD</currency>
        <deliveryMethod>Off-peak</deliveryMethod>
        <usingGeneralRail>N</usingGeneralRail>
        <usingGeneralRoad>Y</usingGeneralRoad>
        <usingGeneralSea>N</usingGeneralSea>
        <usingExpressAir>N</usingExpressAir>
        <usingExpressRoad>N</usingExpressRoad>
        <etaFrom>2</etaFrom>
        <etaTo>3</etaTo>
        <guaranteedEta>Y</guaranteedEta>
        <adjustments>
          <adjustment>
            <description>Credit Card Payment Adjustment</description>
            <amount>2.00</amount>
            <tax>0.20</tax>
          </adjustment>
          <adjustment>
            <description>Credit Payment Adjustment</description>
            <amount>-2.00</amount>
            <tax>-0.20</tax>
          </adjustment>
        </adjustments>
        <inclusions>
          <inclusion>
            <summary>Tracking available</summary>
            <details>Tracking information will be available for this quote.
              </details>
          </inclusion>
          <inclusion>
            <summary>Automatic Booking</summary>
            <details>Automatic booking generated.</details>
          </inclusion>
        </inclusions>
      </quote>
    </ns1:getQuotesByRequestResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

```

    <inclusion>
    <inclusion>
      <summary>Automatic Consignment</summary>
      <details>Automatic consignment note generated.</details>
    </inclusion>
  </inclusions>
  <extras>
    <extra>
      <summary>Insurance</summary>
      <details>Insurance cover up to $5,000.</details>
      <totalPrice>22.00</totalPrice>
      <basePrice>20.00</basePrice>
      <tax>2.00</tax>
      <mandatory>Y</mandatory>
      <adjustments>
        <adjustment>
          <description>Credit Card Payment Adjustment</description>
          <amount>0.40</amount>
          <tax>0.04</tax>
        </adjustment>
      </adjustments>
    </extra>
  </extras>
  <carrier>
    <id>654321</id>
    <companyName>ZYX Transport</companyName>
    <companyContact>John Doe</companyContact>
    <streetAddress>1 Transport Street</streetAddress>
    <streetSuburb>Brisbane</streetSuburb>
    <streetCity>Brisbane</streetCity>
    <streetState>QLD</streetState>
    <streetCode>4000</streetCode>
    <streetCountry>AU</streetCountry>
    <postalAddress>PO Box 1111</postalAddress>
    <postalSuburb>Brisbane</postalSuburb>
    <postalCity>Brisbane</postalCity>
    <postalState>QLD</postalState>
    <postalCode>4001</postalCode>
    <postalCountry>AU</postalCountry>
    <phone1>(07) 3333 3333</phone1>
    <email>john.doe@zyxtransport.com.au</email>
    <website>https://www.zyxtransport.com.au</website>
    <conditions>Our conditions include: lots of text.</conditions>
  </carrier>
</quote>
</ns1:getQuotesByRequestResponse>
</soap-env:Body>
</soap-env:Envelope>

```

2.6.2. makeBookingByRequest

The makeBookingByRequest operation creates a request entry in the system, creates and associates a quote and processes a payment.

The response message is a makeBookingByRequestResponse, and contains a request id.

If a query contains invalid data or payment is unable to be processed then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.2.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.2.2. Request – makeBookingByRequest

```
<xsd:element name="makeBookingByRequest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="anythings">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="anything" type="com:Anything" maxOccurs="10"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="anywhere" type="com:Anywhere"/>
      <xsd:element name="anytime" type="com:Anytime" minOccurs="0"/>
      <xsd:element name="general" type="com:General" minOccurs="0"/>
      <xsd:element name="origin" type="com:Location"/>
      <xsd:element name="destination" type="com:Location"/>
      <xsd:element name="quote" type="com:BookingQuote"/>
      <xsd:element name="payment" type="com:Payment"/>
      <xsd:element name="instructions" type="com:Instructions"
        minOccurs="0"/>
      <xsd:element name="comments" type="com:Comments" minOccurs="0"/>
      <xsd:element name="reference" type="com:ClientReference"
        minOccurs="0"/>
      <xsd:element name="promotionCode" type="com:PromotionCode"
        minOccurs="0"/>
      <xsd:element name="clientId" type="com:ClientId"
        minOccurs="0"/>
      <xsd:element name="labelPrinterType" type="com:LabelPrinterType"
        minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details required to lodge a request and an accepted quote. Also contains the details required for payment processing.

Field	Base type	Occurs	Description
anythings	list	Always	The details about the item/s

			being sent. See section 2.8.1.2 Anything for further details.
anywhere	complex	Always	The details about where the item/s are being sent from/to. See section 2.8.1.3 Anywhere for further details.
anytime	complex	Optional	The details about when the item/s are being sent. If not specified defaults to the next available date. See section 2.8.1.4 Anytime for further details.
general	complex	Optional	The general details about the item/s being sent. See section 2.8.1.5 General for further details.
origin	complex	Always	The details about where an item/s is being sent from or the originating address. See section 2.8.1.12 Location for further details.
destination	complex	Always	The details about where an item/s is being sent to or the final destination address. See section 2.8.1.12 Location for further details.
quote	complex	Always	Specific details about a quote to be accepted. See section 2.8.1.14 BookingQuote for further details.
payment	complex	Always	The details about the payment for the quote. See section 2.8.1.13 Payment for further details.
instructions	simple	Optional	Delivery instructions relating to the request. See section 2.8.2.33 Instructions for further details.
comments	simple	Optional	Client comments relating to the request. See section 2.8.2.34 Comments for further details.
reference	simple	Optional	Client reference relating to the request. See section 2.8.2.102 ClientReference for further

			details.
promotionCode	simple	Optional*	Temando code used for promotions or identifying sales representatives. See section 2.8.2.74 PromotionCode for further details.
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.
labelPrinterType	simple	Optional	The type of printer that will be used for printing the item label/s. If no label/s required this element should not be supplied. See section 2.8.2.103 LabelPrinterType for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
promotionCode	<ul style="list-style-type: none"> Mandatory for users with <i>API (Affiliate): General</i> access rights.
clientId	<ul style="list-style-type: none"> Only applies for users with <i>API (Affiliate): General</i> access rights. Any supplied values are disregarded in all other cases. Mandatory for users with <i>API (Affiliate): General</i> access rights.

Example:

Book the quote for \$110 for XYZ Transport (method Off-peak) for sending a Ford, Mustang, GT Convertible 2dr Auto 3sp 289, 1965 (running but not registered), door-to-door, from a business in Brisbane (4000) (with notifications before pick-up) to a business in Sydney (2000) with pickup on 01 July 2009 in the morning. Payment method is with a credit card.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <body>
    <quote>
      <method>Off-peak</method>
      <from>Brisbane</from>
      <to>Sydney</to>
      <pickup>01 July 2009</pickup>
      <payment>Credit Card</payment>
    </quote>
  </body>
</soapenv:Envelope>
```

```

</wsse:Security>
</soapenv:Header>
<soapenv:Body>
  <tem:makeBookingByRequest
    xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
    <anythings>
      <anything>
        <class>Vehicle</class>
        <subclass>Car</subclass>
        <qualifierVehicleMake>Ford</qualifierVehicleMake>
        <qualifierVehicleModel>Mustang</qualifierVehicleModel>
        <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
          289</qualifierVehicleDescription>
        <qualifierVehicleRunning>Y</qualifierVehicleRunning>
        <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
        <qualifierVehicleYear>1965</qualifierVehicleYear>
        <quantity>1</quantity>
      </anything>
    </anythings>
    <anywhere>
      <itemNature>Domestic</itemNature>
      <itemMethod>Door to Door</itemMethod>
      <originCountry>AU</originCountry>
      <originCode>4000</originCode>
      <originSuburb>Brisbane</originSuburb>
      <destinationCountry>AU</destinationCountry>
      <destinationCode>2000</destinationCode>
      <destinationSuburb>Sydney</destinationSuburb>
      <destinationIs>Residence</destinationIs>
      <destinationResPostalBox>N</destinationResPostalBox>
      <destinationResUnattended>N</destinationResUnattended>
      <destinationResInside>N</destinationResInside>
      <destinationResLimitedAccess>N</destinationResLimitedAccess>
      <destinationResHeavyLift>N</destinationResHeavyLift>
      <originIs>Business</originIs>
      <originBusUnattended>N</originBusUnattended>
      <originBusDock>Y</originBusDock>
      <originBusForklift>N</originBusForklift>
      <originBusLoadingFacilities>N</originBusLoadingFacilities>
      <originBusInside>N</originBusInside>
      <originBusLimitedAccess>N</originBusLimitedAccess>
      <originBusHeavyLift>N</originBusHeavyLift>
      <originBusTailgateLifter>N</originBusTailgateLifter>
      <originBusContainerSwingLifter>N</originBusContainerSwingLifter>
    </anywhere>
    <anytime>
      <readyDate>2009-07-01</readyDate>
      <readyTime>PM</readyTime>
    </anytime>
    <general>
      <goodsValue>2000.00</goodsValue>
    </general>
    <origin>
      <description>ABC Suppliers</description>
      <contactName>John Doe</contactName>
      <companyName>ABC Suppliers</companyName>
      <street>1 Originating Street</street>
      <suburb>Brisbane</suburb>
      <state>QLD</state>
      <code>4000</code>
      <country>AU</country>
      <phone1>(07) 3333 3333</phone1>
      <phone2></phone2>
      <fax>(07) 3333 3331</fax>
      <email>bneoffice@abcsuppliers.com.au</email>
    </origin>
    <destination>
      <contactName>Jane Doe</contactName>
      <street>2 Recipient Street</street>

```

```

    <suburb>Sydney</suburb>
    <state>NSW</state>
    <code>2000</code>
    <country>AU</country>
    <phone1>(02) 9333 3333</phone1>
    <phone2></phone2>
    <fax>(02) 9333 3331</fax>
    <email>jane.doe@yahoo.com.au</email>
  </destination>
  <quote>
    <totalPrice>110.00</totalPrice>
    <basePrice>100.00</basePrice>
    <tax>10.00</tax>
    <currency>AUD</currency>
    <deliveryMethod>Off-peak</deliveryMethod>
    <etaFrom>2</etaFrom>
    <etaTo>3</etaTo>
    <guaranteedEta>Y</guaranteedEta>
    <carrierId>654321</carrierId>
    <extras>
      <extra>
        <summary>Insurance</summary>
        <totalPrice>22.00</totalPrice>
        <basePrice>20.00</basePrice>
        <tax>2.00</tax>
      </extra>
    </extras>
  </quote>
  <payment>
    <paymentType>Credit Card</paymentType>
    <cardType>MasterCard</cardType>
    <cardExpiryDate>05-2010</cardExpiryDate>
    <cardNumber>5163190000000000</cardNumber>
    <cardName>JANE DOE</cardName>
  </payment>
  <instructions>Please use side door.</instructions>
  <reference>ABC200000</reference>
  <comments>Please don't use any chains on the axle.</comments>
  <labelPrinterType>Thermal</labelPrinterType>
</tem:makeBookingByRequest>
</soapenv:Body>
</soapenv:Envelope>

```

2.6.2.3. Response - makeBookingByRequestResponse

```

<xsd:element name="makeBookingByRequestResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requestId" type="xsd:positiveInteger"/>
      <xsd:element name="bookingNumber" type="com:BookingNumber"
        minOccurs="0"/>
      <xsd:element name="consignmentNumber" type="com:ConsignmentNumber"
        minOccurs="0"/>
      <xsd:element name="consignmentDocument" type="com:ConsignmentDocument"
        minOccurs="0"/>
      <xsd:element name="consignmentDocumentType"
        type="com:ConsignmentDocumentType" minOccurs="0"/>
      <xsd:element name="labelDocument" type="com:LabelDocument"
        minOccurs="0"/>
      <xsd:element name="labelDocumentType"
        type="com:LabelDocumentType" minOccurs="0"/>
      <xsd:element name="anytime" type="com:Anytime" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

```

Child Content:

Field	Type	Occurs	Description
requestId	positiveInteger	Always	The request id of the created request.
bookingNumber	simple	Optional	The carrier's booking number for the request. This is optional as the carrier may be required to manually entering this reference. See section 2.8.2.75 BookingNumber for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.
consignmentNumber	simple	Optional	The carrier's consignment number identifies a logical grouping of goods that has been consigned. This is optional as the carrier may be required to manually entering this reference. See section 2.8.2.76 ConsignmentNumber for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.
consignmentDocument	simple	Optional	The carrier's consignment document. This is optional as the carrier may not provide a document. <i>Important: The contents of this field are encoded using base64. This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer. When creating a file from this field's contents you will need to ensure that you decode using base64.</i> See section 2.8.2.77 ConsignmentDocument for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.
consignmentDocumentType	simple	Optional	The type of the carrier's consignment document. This is only provided if an entry for the consignmentDocument field is

			<p>provided. See section 2.8.2.78 ConsignmentDocumentType for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.</p>
labelDocument	simple	Optional	<p>The carrier's label document. This is optional as the carrier may not provide a document. <i>Important: The contents of this field are encoded using base64. This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer. When creating a file from this field's contents you will need to ensure that you decode using base64.</i> See section 2.8.2.81 LabelDocument for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.</p>
labelDocumentType	simple	Optional	<p>The type of the carrier's label document. This is only provided if an entry for the labelDocument field is provided. See section 2.8.2.82 LabelDocumentType for further details. Can be retrieve at a later date via the 2.6.3 getRequest call.</p>
anytime	complex	Optional	<p>The details about when the item/s are being sent. Only returned if an anytime element was not supplied in the request. See section 2.8.1.4 Anytime for further details.</p>

Example:

The request id and booking information for the created request.

```
<soap-env:Envelope
  xmlns:soap-env=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:makeBookingByRequestResponse
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
```

```
<requestId>100000</requestId>
<bookingNumber>XYZ300001</bookingNumber>
<consignmentNumber>UVW400001</consignmentNumber>
<consignmentDocument>JVBER ... shortened for readability ... XRhbg
  </consignmentDocument>
<consignmentDocumentType>application/pdf</consignmentDocumentType>
<labelDocument>e3Rf3 ... shortened for readability ... X8ggy
  </labelDocument>
<labelDocumentType>application/pdf</labelDocumentType>
</ns1:makeBookingByRequestResponse>
</soap-env:Body>
</soap-env:Envelope>
```

2.6.3. getRequest

The getRequest operation retrieves a request and associated details based on the supplied request id.

The response message is a getRequestResponse, and contains all information relating to a request.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.3.1. Access Rights

Only available to users with:

- API (Client): General
- API (Carrier): General
- API (Affiliate): General

2.6.3.2. Request – getRequest

```
<xsd:element name="getRequest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requestId" type="xsd:positiveInteger"
        minOccurs="0"/>
      <xsd:element name="bookingNumber" type="com:BookingNumber"
        minOccurs="0"/>
      <xsd:element name="reference" type="com:ClientReference"
        minOccurs="0"/>
      <xsd:element name="detail" type="com:Detail"
        minOccurs="0"/>
      <xsd:element name="labelPrinterType" type="com:LabelPrinterType"
        minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

Field	Type	Occurs	Description
requestId	positiveInteger	Optional*	The request id of the request required.
bookingNumber	simple	Optional*	The carrier's booking number for the request. See section 2.8.2.75 BookingNumber for further details.
reference	simple	Optional*	Client reference relating to the request. See section 2.8.2.102 ClientReference for further details.

detail	simple	Optional	Specifies the amount of detail required in the returned response. If value is: <ul style="list-style-type: none"> Summary: the consignment and label documents will not be returned. Detailed: all elements returned. If no value supplied will default to "Summary". See section 2.8.2.104 Detail for further details.
labelPrinterType	simple	Optional	The type of printer that will be used for printing the item label/s. If no label/s required this element should not be supplied. Will only be used if supplied detail value is Detailed. See section 2.8.2.103 LabelPrinterType for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
requestId	<ul style="list-style-type: none"> Mandatory if a reference is not supplied and user has <i>API (Client): General</i> and <i>API (Affiliate): General</i> access rights. Mandatory if a bookingNumber is not supplied and user has <i>API (Carrier): General</i> access rights.
bookingNumber	<ul style="list-style-type: none"> Only applies for users with <i>API (Carrier): General</i> access rights. Any supplied values are disregarded in all other cases. Mandatory if a requestId is not supplied and user has <i>API (Carrier): General</i> access rights.
reference	<ul style="list-style-type: none"> Mandatory if a requestId is not supplied and user has <i>API (Client): General</i> and <i>API (Affiliate): General</i> access rights.

Example:

Retrieve request details for the request with the id of "100000".

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```



```

    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <wsse:Security
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
    <wsse:UsernameToken>
      <wsse:Username>username</wsse:Username>
      <wsse:Password>password</wsse:Password>
    </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
<soapenv:Body>
  <tem:getRequest
    xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
    <requestId>100000</requestId>
    <detail>Detailed</detail>
    <labelPrinterType>Thermal</labelPrinterType>
  </tem:getRequest>
</soapenv:Body>
</soapenv:Envelope>

```

2.6.3.3. Response - getRequestResponse

```

<xsd:element name="getRequestResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="request" type="com:Request"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

```

Child Content:

The details relating to a lodged request and associated quote/s.

Field	Base type	Occurs	Description
Request	complex	Always	The specific details for a request. See section 2.8.1.1 Request for further details.

Example:

The request with the id of “100000”.

```

<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getRequestResponse id="100000"
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
      <status>Completed</status>
      <anythings>
        <anything>
          <class>Vehicle</class>
          <subclass>Car</subclass>
          <qualifierVehicleMake>Ford</qualifierVehicleMake>
          <qualifierVehicleModel>Mustang</qualifierVehicleModel>
          <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
            289</qualifierVehicleDescription>

```

```

    <qualifierVehicleRunning>Y</qualifierVehicleRunning>
    <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
    <qualifierVehicleYear>1965</qualifierVehicleYear>
    <quantity>1</quantity>
  </anything>
</anythings>
<anywhere>
  <itemNature>Domestic</itemNature>
  <itemMethod>Door to Door</itemMethod>
  <originCountry>AU</originCountry>
  <originCode>4000</originCode>
  <originSuburb>Brisbane</originSuburb>
  <destinationCountry>AU</destinationCountry>
  <destinationCode>2000</destinationCode>
  <destinationSuburb>Sydney</destinationSuburb>
  <destinationIs>Residence</destinationIs>
  <destinationResPostalBox>N</destinationResPostalBox>
  <destinationResUnattended>N</destinationResUnattended>
  <destinationResInside>N</destinationResInside>
  <destinationResLimitedAccess>N</destinationResLimitedAccess>
  <destinationResHeavyLift>N</destinationResHeavyLift>
  <originIs>Business</originIs>
  <originBusUnattended>N</originBusUnattended>
  <originBusDock>Y</originBusDock>
  <originBusForklift>N</originBusForklift>
  <originBusLoadingFacilities>N</originBusLoadingFacilities>
  <originBusInside>N</originBusInside>
  <originBusLimitedAccess>N</originBusLimitedAccess>
  <originBusHeavyLift>N</originBusHeavyLift>
  <originBusTailgateLifter>N</originBusTailgateLifter>
  <originBusContainersSwingLifter>N</originBusContainersSwingLifter>
</anywhere>
<anytime>
  <readyDate>2009-07-01</readyDate>
  <readyTime>PM</readyTime>
</anytime>
<general>
  <goodsValue>2000.00</goodsValue>
</general>
<origin>
  <description>ABC Suppliers</description>
  <contactName>John Doe</contactName>
  <companyName>ABC Suppliers</companyName>
  <street>1 Originating Street</street>
  <suburb>Brisbane</suburb>
  <state>QLD</state>
  <code>4000</code>
  <country>AU</country>
  <phone1>(07) 3333 3333</phone1>
  <phone2></phone2>
  <fax>(07) 3333 3331</fax>
  <email>bneoffice@abcsuppliers.com.au</email>
</origin>
<destination>
  <contactName>Jane Doe</contactName>
  <street>2 Recipient Street</street>
  <suburb>Sydney</suburb>
  <state>NSW</state>
  <code>2000</code>
  <country>AU</country>
  <phone1>(02) 9333 3333</phone1>
  <phone2></phone2>
  <fax>(02) 9333 3331</fax>
  <email>jane.doe@yahoo.com.au</email>
</destination>
<quotes>
  <quote quote id="111111">
    <generated>Automatically</generated>
    <accepted>Y</accepted>
  </quote>
</quotes>

```

```

<bookingNumber>XYZ300001</bookingNumber>
<consignmentNumber>UVW400001</consignmentNumber>
<consignmentDocument>JVBER ... shortened for readability ... XRhbg
  </consignmentDocument>
<consignmentDocumentType>application/pdf</consignmentDocumentType>
<labelDocument>e3Rf3 ... shortened for readability ... X8ggy
  </labelDocument>
<labelDocumentType>application/pdf</labelDocumentType>
<trackingStatus>In Transit</trackingStatus>
<trackingStatusOccurred>2009-07-02T15:25:46Z
  </trackingStatusOccurred>
<trackingLastChecked>2009-07-02T18:13:12Z</trackingLastChecked>
<trackingFurtherDetails></trackingFurtherDetails>
<totalPrice>110.00</totalPrice>
<basePrice>100.00</basePrice>
<tax>10.00</tax>
<currency>AUD</currency>
<deliveryMethod>Off-peak</deliveryMethod>
<usingGeneralRail>N</usingGeneralRail>
<usingGeneralRoad>Y</usingGeneralRoad>
<usingGeneralSea>N</usingGeneralSea>
<usingExpressAir>N</usingExpressAir>
<usingExpressRoad>N</usingExpressRoad>
<etaFrom>2</etaFrom>
<etaTo>3</etaTo>
<guaranteedEta>Y</guaranteedEta>
<adjustments>
  <adjustment>
    <description>Credit Card Payment Adjustment</description>
    <amount>2.00</amount>
    <tax>0.20</tax>
  </adjustment>
  <adjustment>
    <description>Credit Payment Adjustment</description>
    <amount>-2.00</amount>
    <tax>-0.20</tax>
  </adjustment>
</adjustments>
<inclusions>
  <inclusion>
    <summary>Tracking available</summary>
    <details>Tracking information will be available for this quote.
      </details>
  </inclusion>
  <inclusion>
    <summary>Automatic Booking</summary>
    <details>Automatic booking generated.</details>
  </inclusion>
  <inclusion>
    <summary>Automatic Consignment</summary>
    <details>Automatic consignment note generated.</details>
  </inclusion>
</inclusions>
<extras>
  <extra>
    <summary>Insurance</summary>
    <details>Insurance cover up to $5,000.</details>
    <totalPrice>22.00</totalPrice>
    <basePrice>20.00</basePrice>
    <tax>2.00</tax>
    <mandatory>Y</mandatory>
    <adjustments>
      <adjustment>
        <description>Credit Card Payment Adjustment</description>
        <amount>0.40</amount>
        <tax>0.04</tax>
      </adjustment>
    </adjustments>
  </extra>
</extras>

```

```
</extras>
<carrier>
  <id>654321</id>
  <companyName>ZYX Transport</companyName>
  <companyContact>John Doe</companyContact>
  <streetAddress>1 Transport Street</streetAddress>
  <streetSuburb>Brisbane</streetSuburb>
  <streetCity>Brisbane</streetCity>
  <streetState>QLD</streetState>
  <streetCode>4000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 1111</postalAddress>
  <postalSuburb>Brisbane</postalSuburb>
  <postalCity>Brisbane</postalCity>
  <postalState>QLD</postalState>
  <postalCode>4001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(07) 3333 3333</phone1>
  <email>john.doe@zyxtransport.com.au</email>
  <website>https://www.zyxtransport.com.au</website>
  <conditions>Our conditions include: lots of text.</conditions>
</carrier>
</quote>
</quotes>
<instructions>Please use side door.</instructions>
<reference>ABC200000</reference>
<comments>Please don't use any chains on the axle.</comments>
</ns1:getRequestResponse>
</soap-env:Body>
</soap-env:Envelope>
```

2.6.4. getRequestsRequiringBooking

The getRequestsRequiringBooking operation retrieves a list of all request and associated details assigned to the calling carrier, which have not been booked.

The response message is a getRequestsRequiringBookingResponse, and contains a list of all the requests.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.4.1. Access Rights

Only available to users with:

- API (Carrier): General

2.6.4.2. Request – getRequestsRequiringBooking

No request details are required. Carrier will be identified from login details.

2.6.4.3. Response - getRequestsRequiringBookingResponse

```
<xsd:element name="getRequestsRequiringBookingResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="request" type="com:Request"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details relating to a lodged request and associated quote/s.

Field	Base type	Occurs	Description
request	list	Always	The specific details for all unbooked requests. See section 2.8.1.1 Request for further details.

Example:

List of all the unbooked requests associated with the user with the username “username”.

```
<soap-env:Envelope
  xmlns:soap-env=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getRequestsRequiringBookingResponse
```

```

    xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
<request id="100000">
  <status>Awaiting Booking</status>
  <anythings>
    <anything>
      <class>Vehicle</class>
      <subclass>Car</subclass>
      <qualifierVehicleMake>Ford</qualifierVehicleMake>
      <qualifierVehicleModel>Mustang</qualifierVehicleModel>
      <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
        289</qualifierVehicleDescription>
      <qualifierVehicleRunning>Y</qualifierVehicleRunning>
      <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
      <qualifierVehicleYear>1965</qualifierVehicleYear>
      <quantity>1</quantity>
    </anything>
  </anythings>
  <anywhere>
    <itemNature>Domestic</itemNature>
    <itemMethod>Door to Door</itemMethod>
    <originCountry>AU</originCountry>
    <originCode>4000</originCode>
    <originSuburb>Brisbane</originSuburb>
    <destinationCountry>AU</destinationCountry>
    <destinationCode>2000</destinationCode>
    <destinationSuburb>Sydney</destinationSuburb>
    <destinationIs>Residence</destinationIs>
    <destinationResPostalBox>N</destinationResPostalBox>
    <destinationResUnattended>N</destinationResUnattended>
    <destinationResInside>N</destinationResInside>
    <destinationResLimitedAccess>N</destinationResLimitedAccess>
    <destinationResHeavyLift>N</destinationResHeavyLift>
    <originIs>Business</originIs>
    <originBusUnattended>N</originBusUnattended>
    <originBusDock>Y</originBusDock>
    <originBusForklift>N</originBusForklift>
    <originBusLoadingFacilities>N</originBusLoadingFacilities>
    <originBusInside>N</originBusInside>
    <originBusLimitedAccess>N</originBusLimitedAccess>
    <originBusHeavyLift>N</originBusHeavyLift>
    <originBusTailgateLifter>N</originBusTailgateLifter>
    <originBusContainersSwingLifter>N</originBusContainersSwingLifter>
  </anywhere>
  <anytime>
    <readyDate>2009-07-01</readyDate>
    <readyTime>PM</readyTime>
  </anytime>
  <general>
    <goodsValue>2000.00</goodsValue>
  </general>
  <origin>
    <description>ABC Suppliers</description>
    <contactName>John Doe</contactName>
    <companyName>ABC Suppliers</companyName>
    <street>1 Originating Street</street>
    <suburb>Brisbane</suburb>
    <state>QLD</state>
    <code>4000</code>
    <country>AU</country>
    <phone1>(07) 3333 3333</phone1>
    <phone2></phone2>
    <fax>(07) 3333 3331</fax>
    <email>bneoffice@abcsuppliers.com.au</email>
  </origin>
  <destination>
    <contactName>Jane Doe</contactName>
    <street>2 Recipient Street</street>
    <suburb>Sydney</suburb>
    <state>NSW</state>
  </destination>
</request>

```

```

<code>2000</code>
<country>AU</country>
<phone1>(02) 9333 3333</phone1>
<phone2></phone2>
<fax>(02) 9333 3331</fax>
<email>jane.doe@yahoo.com.au</email>
</destination>
<quotes>
  <quote id="111111">
    <generated>Automatically</generated>
    <accepted>Y</accepted>
    <totalPrice>110.00</totalPrice>
    <basePrice>100.00</basePrice>
    <tax>10.00</tax>
    <currency>AUD</currency>
    <deliveryMethod>Off-peak</deliveryMethod>
    <usingGeneralRail>N</usingGeneralRail>
    <usingGeneralRoad>Y</usingGeneralRoad>
    <usingGeneralSea>N</usingGeneralSea>
    <usingExpressAir>N</usingExpressAir>
    <usingExpressRoad>N</usingExpressRoad>
    <etaFrom>2</etaFrom>
    <etaTo>3</etaTo>
    <guaranteedEta>Y</guaranteedEta>
    <adjustments/>
    <inclusions>
      <inclusion>
        <summary>Tracking available</summary>
        <details>Tracking information will be available for this
          quote.</details>
      </inclusion>
      <inclusion>
        <summary>Automatic Booking</summary>
        <details>Automatic booking generated.</details>
      </inclusion>
      <inclusion>
        <summary>Automatic Consignment</summary>
        <details>Automatic consignment note generated.</details>
      </inclusion>
    </inclusions>
    <extras/>
    <carrier>
      <id>654321</id>
      <companyName>ZYX Transport</companyName>
      <companyContact>John Doe</companyContact>
      <streetAddress>1 Transport Street</streetAddress>
      <streetSuburb>Brisbane</streetSuburb>
      <streetCity>Brisbane</streetCity>
      <streetState>QLD</streetState>
      <streetCode>4000</streetCode>
      <streetCountry>AU</streetCountry>
      <postalAddress>PO Box 1111</postalAddress>
      <postalSuburb>Brisbane</postalSuburb>
      <postalCity>Brisbane</postalCity>
      <postalState>QLD</postalState>
      <postalCode>4001</postalCode>
      <postalCountry>AU</postalCountry>
      <phone1>(07) 3333 3333</phone1>
      <email>john.doe@zyxtransport.com.au</email>
      <website>https://www.zyxtransport.com.au</website>
      <conditions>Our conditions include: lots of text.</conditions>
    </carrier>
  </quote>
</quotes>
<instructions>Please use side door.</instructions>
<reference>ABC200000</reference>
<comments>Please don't use any chains on the axle.</comments>
</request>
</ns1:getRequestsRequiringBookingResponse>

```

```
</soap-env:Body>  
</soap-env:Envelope>
```


2.6.5. addBookingDetails

The addBookingDetails operation adds the booking information (eg. booking number, consignment number, consignment file, etc) to the supplied request id.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.5.1. Access Rights

Only available to users with:

- API (Carrier): General

2.6.5.2. Request – addBookingDetails

```
<xsd:element name="addBookingDetails">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requestId" type="xsd:positiveInteger"/>
      <xsd:element name="bookingNumber" type="com:BookingNumber"/>
      <xsd:element name="consignmentNumber" type="com:ConsignmentNumber"
        minOccurs="0"/>
      <xsd:element name="consignmentDocument"
        type="com:ConsignmentDocument" minOccurs="0"/>
      <xsd:element name="consignmentDocumentType"
        type="com:ConsignmentDocumentType" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

Field	Type	Occurs	Description
requestId	positiveInteger	Always	The request id of the request required.
bookingNumber	simple	Always	The carrier booking number for the request. See section 2.8.2.75 BookingNumber for further details.
consignmentNumber	simple	Optional	The carrier consignment number for the request. If excluded will reference the bookingNumber. See section 2.8.2.76 ConsignmentNumber for further details.
consignmentDocument	simple	Optional	The carrier consignment

			<p>document.</p> <p>Important: The contents of this field must be encoded using base64.</p> <p>This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.</p> <p>See section 2.8.2.77 ConsignmentDocument for further details.</p>
consignmentDocumentType	simple	Optional	<p>The type of the carrier consignment document. This is only required if an entry for the consignmentDocument field is provided.</p> <p>See section 2.8.2.78 ConsignmentDocumentType for further details.</p>

Example:

Assign the booking reference of “XYZ300001” to the request with the id of “100000”. Also send a pdf document of the consignment note.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
      <tem:addBookingDetails
        xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
        <requestId>100000</requestId>
        <bookingNumber>XYZ300001</bookingNumber>
        <consignmentNumber>UVW400001</consignmentNumber>
        <consignmentDocument>JVBER ... shortened for readability ... XRhbg
          </consignmentDocument>
        <consignmentDocumentType>application/pdf</consignmentDocumentType>
      </tem:addBookingDetails>
    </soapenv:Body>
  </soapenv:Envelope>
```

2.6.5.3. Response – addBookingDetailsResponse

No response will be returned.

2.6.6. updateTrackingDetails

The updateTrackingDetails operation updates the tracking information (eg. tracking status, when status occurred, etc) to the supplied request id.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.6.1. Access Rights

Only available to users with:

- API (Carrier): General

2.6.6.2. Request – updateTrackingDetails

```
<xsd:element name="updateTrackingDetails">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requestId" type="xsd:positiveInteger"
        minOccurs="0"/>
      <xsd:element name="bookingNumber" type="com:BookingNumber"
        minOccurs="0"/>
      <xsd:element name="trackingStatus" type="com:TrackingStatus"/>
      <xsd:element name="trackingStatusOccurred" type="com:Datetime"/>
      <xsd:element name="trackingFurtherDetails"
        type="com:TrackingFurtherDetails" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

Field	Type	Occurs	Description
requestId	positiveInteger	Optional*	The request id of the request required.
bookingNumber	simple	Optional*	The carrier's booking number for the request. See section 2.8.2.75 BookingNumber for further details.
trackingStatus	simple	Always	The status of the transit of the item/s being sent. See section 2.8.2.83 TrackingStatus for further details.
trackingStatusOccurred	simple	Always	When the tracking status event occurred. See section 2.8.2.51 Datetime for further details.
trackingFurtherDetails	simple	Optional	Further details on the transit of the item/s being sent. See section 2.8.2.84 TrackingFurtherDetails for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
requestId	<ul style="list-style-type: none"> Mandatory for users with <i>API (Client): General</i> and <i>API (Affiliate): General</i> access rights. Mandatory if a bookingNumber is not supplied and user has <i>API (Carrier): General</i> access rights.
bookingNumber	<ul style="list-style-type: none"> Only applies for users with <i>API (Carrier): General</i> access rights. Any supplied values are disregarded in all other cases. Mandatory if a requestId is not supplied and user has <i>API (Carrier): General</i> access rights.

Example:

Assign the tracking status of "Delivered" with a status occurred datetime at 9:35:23am, 3rd July 2009 (UTC) to the request with the id of "100000".

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
      </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
      <tem:updateTrackingDetails
        xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
        <requestId>100000</requestId>
        <trackingStatus>Delivered</trackingStatus>
        <trackingStatusOccurred>2009-07-03T09:35:23Z
          </trackingStatusOccurred>
        </tem:updateTrackingDetails>
      </soapenv:Body>
    </soapenv:Envelope>
```

2.6.6.3. Response – updateTrackingDetailsResponse

No response will be returned.

2.6.7. createClient

The createClient operation creates a new client account in the system.

The response message is a createClientResponse, and contains a client id.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.7.1. Access Rights

Only available to users with:

- API (Affiliate): General

2.6.7.2. Request – createClient

```
<xsd:element name="createClient">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="loginDetails" type="com:LoginDetails"/>
      <xsd:element name="client" type="com:Client"/>
      <xsd:element name="promotionCode" type="com:PromotionCode"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details required to create a client.

Field	Base type	Occurs	Description
loginDetails	complex	Always	The login details for the client. See section 2.8.1.16 LoginDetails for further details.
client	complex	Always	The details about the client. See section 2.8.1.15 Client for further details.
promotionCode	simple	Always	Temando code used for promotions or identifying sales representatives. See section 2.8.2.74 PromotionCode for further details.

Example:

Lodge a request for creating an account for the individual John Smith with the details of 1 User Street, Melbourne, VIC, 3000, AU (PO Box 3333, Melbourne, VIC, 3001, AU), ph. (03) 9999 9999, john.smith@mailhost.com.au.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
```

```

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
  <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
    <wsse:UsernameToken>
      <wsse:Username>username</wsse:Username>
      <wsse:Password>password</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
<soapenv:Body>
  <tem:createClient
    xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
    <loginDetails>
      <loginId>jsmith</loginId>
      <password>my_password1357</password>
    </loginDetails>
    <client>
      <individualCompany>Individual</individualCompany>
      <individualSurname>Smith</individualSurname>
      <individualFirstname>John</individualFirstname>
      <individualDateOfBirth>1980-04-21</individualDateOfBirth>
      <streetAddress>1 User Street</streetAddress>
      <streetSuburb>Melbourne</streetSuburb>
      <streetCity>Melbourne</streetCity>
      <streetState>VIC</streetState>
      <streetCode>3000</streetCode>
      <streetCountry>AU</streetCountry>
      <postalAddress>PO Box 3333</postalAddress>
      <postalSuburb>Melbourne</postalSuburb>
      <postalCity>Melbourne</postalCity>
      <postalState>VIC</postalState>
      <postalCode>3001</postalCode>
      <postalCountry>AU</postalCountry>
      <phone1>(03) 9999 9999</phone1>
      <email>john.smith@mailhost.com.au</email>
    </client>
    <promotionCode>X0001</promotionCode>
  </tem:createClient>
</soapenv:Body>
</soapenv:Envelope>

```

2.6.7.3. Response - createClientResponse

```

<xsd:element name="createClientResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="clientId" type="com:ClientId"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

```

Child Content:

Field	Type	Occurs	Description
clientId	simple	Always	The client id of the created client. See section 2.8.2.93 ClientId for further details.

Example:

The client id for the created client.

```
<soap-env:Envelope
  xmlns:soap-env=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:createClientResponse
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
      <clientId>123456</clientId>
    </ns1:createClientResponse>
  </soap-env:Body>
</soap-env:Envelope>
```


2.6.8. updateClient

The updateClient operation updates an existing client's account details.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.8.1. Access Rights

Only available to users with:

- API (Affiliate): General

2.6.8.2. Request – updateClient

```
<xsd:element name="updateClient">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="client" type="com:Client"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details required to update a client.

Field	Base type	Occurs	Description
client	complex	Always	The details about the client. See section 2.8.1.15 Client for further details.

Example:

Lodge a request for updating account id 123456 to change the client's surname to Fraser.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:updateClient
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <client>
        <id>123456</id>
        <individualSurname>Fraser</individualSurname>
      </client>
    </tem:updateClient>
```

```
</soapenv:Body>  
</soapenv:Envelope>
```

2.6.8.3. Response – updateClientResponse

No response will be returned.

2.6.9. getClient

The getClient operation retrieves a client's account details based on the supplied client id.

The response message is a getClientResponse, and contains a client's account details.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.9.1. Access Rights

Only available to users with:

- API (Affiliate): General

2.6.9.2. Request – getClient

```
<xsd:element name="getClient">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0">
        <xsd:element name="loginDetails" type="com:LoginDetails"
          minOccurs="0">
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
```

Child Content:

Field	Type	Occurs	Description
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.
loginDetails	complex	Optional*	The login details for the client. Should only be supplied if the client id is unknown. If supplied in conjunction with a clientId then these details (ie. loginDetails) will be disregarded. See section 2.8.1.16 LoginDetails for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
clientId	<ul style="list-style-type: none"> Mandatory if loginDetails is not supplied.
loginDetails	<ul style="list-style-type: none"> Mandatory if clientId is not supplied.

Example: Retrieve account details for the client with the id of “123456”.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:getClient
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <clientId>123456</clientId>
    </tem:getClient>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.9.3. Response - getClientResponse

```
<xsd:element name="getClientResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="client" type="com:Client"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details relating to a client’s account.

Field	Base type	Occurs	Description
client	complex	Always	The details about the client. See section 2.8.1.15 Client for further details.

Example: The account details for the client with the id of “123456”.

```
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getClientResponse id="123456"
      xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
      <client>
        <id>123456</id>
      </client>
    </ns1:getClientResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

```
<individualCompany>Individual</individualCompany>
<individualSurname>Smith</individualSurname>
<individualFirstname>John</individualFirstname>
<individualDateOfBirth>1980-04-21</individualDateOfBirth>
<streetAddress>1 User Street</streetAddress>
<streetSuburb>Melbourne</streetSuburb>
<streetCity>Melbourne</streetCity>
<streetState>VIC</streetState>
<streetCode>3000</streetCode>
<streetCountry>AU</streetCountry>
<postalAddress>PO Box 3333</postalAddress>
<postalSuburb>Melbourne</postalSuburb>
<postalCity>Melbourne</postalCity>
<postalState>VIC</postalState>
<postalCode>3001</postalCode>
<postalCountry>AU</postalCountry>
<phone1>(03) 9999 9999</phone1>
<email>john.smith@mailhost.com.au</email>
</client>
</ns1:getClientResponse>
</soap-env:Body>
</soap-env:Envelope>
```

2.6.10. getManifest

The getManifest operation returns the manifest documents (and label documents, if requested) for all bookings based on the supplied parameters.

The response message is a getManifestResponse, and contains the manifest documents (and label documents, if requested).

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.10.1. Access Rights

Only available to users with:

- API (Client): General
- API (Carrier): General

2.6.10.2. Request – getManifest

```
<xsd:element name="getManifest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="type" type="com:ManifestType" minOccurs="0"/>
      <xsd:element name="carrierId" type="com:CarrierId" minOccurs="0"/>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
      <xsd:element name="location" type="com:Location" minOccurs="0"/>
      <xsd:element name="readyDate" type="com:Date" minOccurs="0"/>
      <xsd:element name="labelPrinterType" type="com:LabelPrinterType"
        minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The dispatch details of the consignment.

Field	Base type	Occurs	Description
type	simple	Optional*	The type of manifest report to be generated. Will default to "Confirmed" if no value is provided. See section 2.6.11 confirmManifest for details on confirming requests. See section 2.8.2.101 ManifestType for further details.
carrierId	simple	Optional*	The carrier's id for returning only those bookings associated with this carrier. See section 2.8.2.58 CarrierId for further details.
clientId	simple	Optional*	The client's id for returning

			only those bookings associated with this client. See section 2.8.2.93 ClientId for further details.
location	simple	Optional	The location description for returning only those bookings associated with this location. See section 2.8.1.12 Location for further details.
readyDate	simple	Optional	The ready date for returning only those bookings with a ready date matching this supplied date. See section 2.8.2.50 Date for further details.
labelPrinterType	simple	Optional	The type of printer that will be used for printing the item label/s. If no label/s required this element should not be supplied. See section 2.8.2.103 LabelPrinterType for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
carrierId	<ul style="list-style-type: none"> Only applies for users with <i>API (Client): General</i> access rights. Any supplied values are disregarded in all other cases.
clientId	<ul style="list-style-type: none"> Only applies for users with <i>API (Carrier): General</i> access rights. Any supplied values are disregarded in all other cases.

Example:

Return manifest and labels (for a Thermal printer) for all bookings being sent from the location ABC Suppliers, on the date 1st July, 2009.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
```

```

    <wsse:UsernameToken>
      <wsse:Username>username</wsse:Username>
      <wsse:Password>password</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
<soapenv:Body>
  <tem:getManifest
    xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
    <type>Confirmed</type>
    <location>ABC Suppliers</location>
    <readyDate>2009-07-01</readyDate>
    <labelPrinterType>Thermal</labelPrinterType>
  </tem:getManifest>
</soapenv:Body>
</soapenv:Envelope>

```

2.6.10.3. Response - getManifestResponse

```

<xsd:element name="getManifestResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requests">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="request" type="com:Request" minOccurs="0"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="manifestDocument" type="com:ManifestDocument"/>
      <xsd:element name="manifestDocumentType"
        type="com:ManifestDocumentType"/>
      <xsd:element name="labelDocument" type="com:LabelDocument"
        minOccurs="0"/>
      <xsd:element name="labelDocumentType" type="com:LabelDocumentType"
        minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

```

Child Content:

The details relating to the associated requests, manifest documents and label documents.

Field	Base type	Occurs	Description
requests	list	Any	A list of corresponding requests. See section 2.8.1.1 Request for further details.
manifestDocument	simple	Always	The manifest document. Important: The contents of this field are encoded using base64. This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.

			<p><i>When creating a file from this field's contents you will need to ensure that you decode using base64.</i></p> <p>See section 2.8.2.79 ManifestDocument for further details.</p>
manifestDocumentType	simple	Always	<p>The type of the manifest document.</p> <p>See section 2.8.2.80 ManifestDocumentType for further details.</p>
labelDocument	simple	Optional	<p>The label document. This will only be provided if a value was provided for the labelPrinterType element.</p> <p><i>Important: The contents of this field are encoded using base64.</i></p> <p><i>This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.</i></p> <p><i>When creating a file from this field's contents you will need to ensure that you decode using base64.</i></p> <p>See section 2.8.2.81 LabelDocument for further details.</p>
labelDocumentType	simple	Optional	<p>The type of the label document. This will only be provided if a value was provided for the labelPrinterType element.</p> <p>See section 2.8.2.82 LabelDocumentType for further details.</p>

Example:

Requests, manifest and labels for all bookings being sent from the location ABC Suppliers, on the 1st July, 2009.

```
<soap-env:Envelope
  xmlns:soap-env=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap-env:Header/>
  <soap-env:Body>
    <ns1:getManifestResponse
```

```

    xmlns:ns1="http://api.temando.com/schema/2009_06/server.xsd">
<requests>
<request id="100000">
<status>Completed</status>
<anythings>
<anything>
<class>Vehicle</class>
<subclass>Car</subclass>
<qualifierVehicleMake>Ford</qualifierVehicleMake>
<qualifierVehicleModel>Mustang</qualifierVehicleModel>
<qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
289</qualifierVehicleDescription>
<qualifierVehicleRunning>Y</qualifierVehicleRunning>
<qualifierVehicleRegistered>N</qualifierVehicleRegistered>
<qualifierVehicleYear>1965</qualifierVehicleYear>
<quantity>1</quantity>
</anything>
</anythings>
<anywhere>
<itemNature>Domestic</itemNature>
<itemMethod>Door to Door</itemMethod>
<originCountry>AU</originCountry>
<originCode>4000</originCode>
<originSuburb>Brisbane</originSuburb>
<destinationCountry>AU</destinationCountry>
<destinationCode>2000</destinationCode>
<destinationSuburb>Sydney</destinationSuburb>
<destinationIs>Residence</destinationIs>
<destinationResPostalBox>N</destinationResPostalBox>
<destinationResUnattended>N</destinationResUnattended>
<destinationResInside>N</destinationResInside>
<destinationResLimitedAccess>N</destinationResLimitedAccess>
<destinationResHeavyLift>N</destinationResHeavyLift>
<originIs>Business</originIs>
<originBusUnattended>N</originBusUnattended>
<originBusDock>Y</originBusDock>
<originBusForklift>N</originBusForklift>
<originBusLoadingFacilities>N</originBusLoadingFacilities>
<originBusInside>N</originBusInside>
<originBusLimitedAccess>N</originBusLimitedAccess>
<originBusHeavyLift>N</originBusHeavyLift>
<originBusTailgateLifter>N</originBusTailgateLifter>
<originBusContainersSwingLifter>N</originBusContainersSwingLifter>
</anywhere>
<anytime>
<readyDate>2009-07-01</readyDate>
<readyTime>PM</readyTime>
</anytime>
<general>
<goodsValue>2000.00</goodsValue>
</general>
<origin>
<description>ABC Suppliers</description>
<contactName>John Doe</contactName>
<companyName>ABC Suppliers</companyName>
<street>1 Originating Street</street>
<suburb>Brisbane</suburb>
<state>QLD</state>
<code>4000</code>
<country>AU</country>
<phone1>(07) 3333 3333</phone1>
<phone2></phone2>
<fax>(07) 3333 3331</fax>
<email>bneoffice@abcsuppliers.com.au</email>
</origin>
<destination>
<contactName>Jane Doe</contactName>
<street>2 Recipient Street</street>
<suburb>Sydney</suburb>

```

```

<state>NSW</state>
<code>2000</code>
<country>AU</country>
<phone1>(02) 9333 3333</phone1>
<phone2></phone2>
<fax>(02) 9333 3331</fax>
<email>jane.doe@yahoo.com.au</email>
</destination>
<quotes>
  <quote quote id="111111">
    <generated>Automatically</generated>
    <accepted>Y</accepted>
    <bookingNumber>XYZ300001</bookingNumber>
    <consignmentNumber>UVW400001</consignmentNumber>
    <trackingStatus>In Transit</trackingStatus>
    <trackingStatusOccurred>2009-07-02T15:25:46Z
      </trackingStatusOccurred>
    <trackingLastChecked>2009-07-02T18:13:12Z</trackingLastChecked>
    <trackingFurtherDetails></trackingFurtherDetails>
    <totalPrice>110.00</totalPrice>
    <basePrice>100.00</basePrice>
    <tax>10.00</tax>
    <currency>AUD</currency>
    <deliveryMethod>Off-peak</deliveryMethod>
    <usingGeneralRail>N</usingGeneralRail>
    <usingGeneralRoad>Y</usingGeneralRoad>
    <usingGeneralSea>N</usingGeneralSea>
    <usingExpressAir>N</usingExpressAir>
    <usingExpressRoad>N</usingExpressRoad>
    <etaFrom>2</etaFrom>
    <etaTo>3</etaTo>
    <guaranteedEta>Y</guaranteedEta>
    <adjustments>
      <adjustment>
        <description>Credit Card Payment Adjustment</description>
        <amount>2.00</amount>
        <tax>0.20</tax>
      </adjustment>
      <adjustment>
        <description>Credit Payment Adjustment</description>
        <amount>-2.00</amount>
        <tax>-0.20</tax>
      </adjustment>
    </adjustments>
    <inclusions>
      <inclusion>
        <summary>Tracking available</summary>
        <details>Tracking information will be available for this quote.
          </details>
      </inclusion>
      <inclusion>
        <summary>Automatic Booking</summary>
        <details>Automatic booking generated.</details>
      </inclusion>
      <inclusion>
        <summary>Automatic Consignment</summary>
        <details>Automatic consignment note generated.</details>
      </inclusion>
    </inclusions>
    <extras>
      <extra>
        <summary>Insurance</summary>
        <details>Insurance cover up to $5,000.</details>
        <totalPrice>22.00</totalPrice>
        <basePrice>20.00</basePrice>
        <tax>2.00</tax>
        <mandatory>Y</mandatory>
      </extra>
    </extras>
  </quote>
</quotes>

```

```

        <description>Credit Card Payment Adjustment</description>
        <amount>0.40</amount>
        <tax>0.04</tax>
      </adjustment>
    </adjustments>
  </extra>
</extras>
<carrier>
  <id>654321</id>
  <companyName>ZYX Transport</companyName>
  <companyContact>John Doe</companyContact>
  <streetAddress>1 Transport Street</streetAddress>
  <streetSuburb>Brisbane</streetSuburb>
  <streetCity>Brisbane</streetCity>
  <streetState>QLD</streetState>
  <streetCode>4000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 1111</postalAddress>
  <postalSuburb>Brisbane</postalSuburb>
  <postalCity>Brisbane</postalCity>
  <postalState>QLD</postalState>
  <postalCode>4001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(07) 3333 3333</phone1>
  <email>john.doe@zyxtransport.com.au</email>
  <website>https://www.zyxtransport.com.au</website>
  <conditions>Our conditions include: lots of text.</conditions>
</carrier>
</quote>
</quotes>
<instructions>Please use side door.</instructions>
<reference>ABC200000</reference>
<comments>Please don't use any chains on the axle.</comments>
</request>
</requests>
<consignmentDocument>JVBER ... shortened for readability ... XRhbg
  </consignmentDocument>
<consignmentDocumentType>application/pdf</consignmentDocumentType>
<labelDocument>e3Rf3 ... shortened for readability ... X8ggy
  </labelDocument>
<labelDocumentType>application/pdf</labelDocumentType>
</ns1:getManifestResponse>
</soap-env:Body>
</soap-env:Envelope>

```

2.6.11. confirmManifest

The confirmManifest operation registers those requests that match the supplied parameters as being confirmed and ready to send. If a confirmedReadyDate is supplied all matching requests are updated to have this new send date.

The response message is a confirmManifestResponse, and contains the manifest documents (and label documents, if requested).

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.11.1. Access Rights

Only available to users with:

- API (Client): General

2.6.11.2. Request – confirmManifest

```
<xsd:element name="confirmManifest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="carrierId" type="com:CarrierId" minOccurs="0"/>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
      <xsd:element name="location" type="com:Location" minOccurs="0"/>
      <xsd:element name="startReadyDate" type="com:Date" minOccurs="0"/>
      <xsd:element name="endReadyDate" type="com:Date" minOccurs="0"/>
      <xsd:element name="confirmedReadyDate" type="com:Date" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The dispatch details of the consignment.

Field	Base type	Occurs	Description
carrierId	simple	Optional	The carrier's id for confirming only those bookings associated with this carrier. See section 2.8.2.58 CarrierId for further details.
location	simple	Optional	The location description for confirming only those bookings associated with this location. See section 2.8.1.12 Location for further details.
startReadyDate	simple	Optional	The start ready date for confirming only those bookings with a ready date after (and including) this

			supplied date. See section 2.8.2.50 Date for further details.
endReadyDate	simple	Optional	The end ready date for confirming only those bookings with a ready date before (and including) this supplied date. See section 2.8.2.50 Date for further details.
confirmedReadyDate	simple	Optional	The ready date that all matching bookings will be updated to. See section 2.8.2.50 Date for further details.

Example:

Confirm all bookings being sent from the location ABC Suppliers, between (and included) the dates 1st July, 2009 and 3rd July, 2009. Also update the send date for all bookings to the 1st July, 2009.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:confirmManifest
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <location>ABC Suppliers</location>
      <startReadyDate>2009-07-01</startReadyDate>
      <endReadyDate>2009-07-03</endReadyDate>
      <confirmedReadyDate>2009-07-01</confirmedReadyDate>
    </tem:confirmManifest>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.11.3. Response – confirmManifestResponse

No response will be returned.

2.6.12. cancelRequest

The cancelRequest operation cancels an existing request entry in the system.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.12.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.12.2. Request – cancelRequest

```
<xsd:element name="cancelRequest">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="requestId" type="xsd:positiveInteger"
        minOccurs="0"/>
      <xsd:element name="reference" type="com:ClientReference"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The request id for the request to be cancelled.

Field	Base type	Occurs	Description
requestId	positiveInteger	Optional*	The request id of the request required.
reference	simple	Optional*	Client reference relating to the request. See section 2.8.2.102 ClientReference for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
requestId	<ul style="list-style-type: none"> • Mandatory if a reference is not supplied.
reference	<ul style="list-style-type: none"> • Mandatory if a requestId is not supplied.

Example:

Cancel the request with the id of “100000”.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:cancelRequest
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <requestId>100000</requestId>
    </tem:cancelRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.12.3. Response – cancelRequestResponse

No response will be returned.

2.6.13. createLocation

The createLocation operation creates a new location in the system.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.13.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.13.2. Request – createClient

```
<xsd:element name="createLocation">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
      <xsd:element name="location" type="com:Location"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details required to create a location.

Field	Base type	Occurs	Description
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.
location	complex	Always	The details about the location. See section 2.8.1.12 Location for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
clientId	<ul style="list-style-type: none"> • Only applies for users with <i>API (Affiliate): General</i> access rights. Any supplied values are disregarded in all other cases. • Mandatory for users with <i>API (Affiliate): General</i> access rights.

Example:

Lodge a request for creating a location with a description of ABC Suppliers and the details of John Doe, Origin, ABC Suppliers, 1 Originating Street, Brisbane, QLD, 4000, AU, ph. (07) 3333 3333, fax. (07) 333 3331, bneoffice@abcsuppliers.com.au.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:createLocation
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <location>
        <description>ABC Suppliers</description>
        <contactName>John Doe</contactName>
        <type>Origin</type>
        <companyName>ABC Suppliers</companyName>
        <street>1 Originating Street</street>
        <suburb>Brisbane</suburb>
        <state>QLD</state>
        <code>4000</code>
        <country>AU</country>
        <phone1>(07) 3333 3333</phone1>
        <fax>(07) 3333 3331</fax>
        <email>bneoffice@abcsuppliers.com.au</email>
      </location>
    </tem:createLocation>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.13.3. Response – createLocationResponse

No response will be returned.

2.6.14. updateLocation

The updateLocation operation updates an existing location's details.

No response message will be returned.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.14.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.14.2. Request – updateLocation

```
<xsd:element name="updateClient">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
      <xsd:element name="location" type="com:Location"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details required to create a location.

Field	Base type	Occurs	Description
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.
location	complex	Always	The details about the location. See section 2.8.1.12 Location for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
clientId	<ul style="list-style-type: none"> • Only applies for users with <i>API (Affiliate): General</i> access rights. Any supplied values are disregarded in all other cases. • Mandatory for users with <i>API (Affiliate): General</i> access rights.

Example:

Lodge a request for updating location with description of ABC Suppliers to change the location's contact name to Joseph Doe.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:updateLocation
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <location>
        <description>ABC Suppliers</description>
        <type>Origin</type>
        <contactName>Joseph Doe</contactName>
      </location>
    </tem:updateLocation>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.14.3. Response – updateLocationResponse

No response will be returned.

2.6.15. getLocation

The getLocation operation retrieves a list of location details based on the supplied parameters.

The response message is a getLocationResponse, and contains a list of location details.

If a query contains invalid data then a SoapFault is thrown with the details of the exception (see section 2.7 SOAP Faults).

2.6.15.1. Access Rights

Only available to users with:

- API (Client): General
- API (Affiliate): General

2.6.15.2. Request – getLocation

```
<xsd:element name="getClient">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="clientId" type="com:ClientId" minOccurs="0"/>
      <xsd:element name="description" type="com:LocationName"
        minOccurs="0"/>
      <xsd:element name="type" type="com:LocationPosition" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

Field	Type	Occurs	Description
clientId	simple	Optional*	The client id of the client's account. See section 2.8.2.93 ClientId for further details.
description	simple	Optional	The description of the location. See section 2.8.2.60 LocationName for further details.
type	simple	Optional	The type of the location. See section 2.8.2.100 LocationPosition for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
clientId	<ul style="list-style-type: none"> Only applies for users with <i>API (Affiliate): General</i> access rights. Any supplied values are disregarded in all other cases. Mandatory for users with <i>API (Affiliate): General</i> access rights.

Example: Retrieve locations with a type of Origin.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password>password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <tem:getLocations
      xmlns:tem="http://api.temando.com/schema/2009_06/server.xsd">
      <type>Origin</type>
    </tem:getLocations>
  </soapenv:Body>
</soapenv:Envelope>
```

2.6.15.3. Response - getLocationResponse

```
<xsd:element name="getClientResponse">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="locations">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="location" type="com:Location" minOccurs="0"
              maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Child Content:

The details of the matching locations.

Field	Base type	Occurs	Description
locations	list	Always	The details about the locations. See section 2.8.1.12 Location for further details.

Example: The list of locations with a type of Origin.

```
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap-env:Header/>
<soap-env:Body>
  <ns1:getLocationsResponse
    xmlns:ns1="http://api.temando.com/schema/2009\_06/server.xsd">
    <locations>
      <location>
        <description>ABC Suppliers</description>
        <contactName>John Doe</contactName>
        <companyName>ABC Suppliers</companyName>
        <street>1 Originating Street</street>
        <suburb>Brisbane</suburb>
        <state>QLD</state>
        <code>4000</code>
        <country>AU</country>
        <phone1>(07) 3333 3333</phone1>
        <phone2></phone2>
        <fax>(07) 3333 3331</fax>
        <email>bneoffice@abcsuppliers.com.au</email>
      </location>
    </locations>
  </ns1:getLocationsResponse>
</soap-env:Body>
</soap-env:Envelope>
```

2.7. SOAP Faults

The following faults are generic (ie. can be returned from any operation) unless specified otherwise. In all cases the faultactor will be set to the accessed URL, and the faultstring will contain a human-readable description of the fault's cause.

- Err_Service:** Fault codes starting with "Err_Service" indicate an issue with the Temando API service.
- Err_Connection:** Fault codes starting with "Err_Connection" indicate an issue with the connection or data transfer method used to communication with the Temando API service.
- Err_Access:** Fault codes starting with "Err_Access" indicate an issue with a user's access rights to the Temando API service and its operations.
- Err_Format:** Fault codes starting with "Err_Format" indicate an issue with the format of the data being sent to the Temando API service.
- Err_Payment:** Fault codes starting with "Err_Payment" indicate an issue with the payment being processed by the Temando API service.
- Err_Permission:** Fault codes starting with "Err_Permission" indicate an issue with the user's permission to access certain data via the Temando API service.
- Err_State:** Fault codes starting with "Err_State" indicate an issue with the user attempting to perform an operation on data that does not exist or is not in the correct state for process.

In all cases a HTTP response code of 500 is returned with the fault.

2.7.1. Err_Service01

This fault is generated when the Temando API service is unavailable. This could occur during an upgrade.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Service01</faultcode>
      <faultstring>The API service is currently unavailable. Please try
        again later.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
```



```
</soap-env:Envelope>
```

2.7.2. Err_Service02

This fault is generated when the Temando API service has encountered an unknown error whilst processing the call. Details of the specific area of failure is detailed in the *<faultstring>* element (to assist Temando with troubleshooting).

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Service02</faultcode>
      <faultstring>The API service is currently having problems processing
        your request - during the authentication process. Please try
        again later.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.3. Err_Service03

This fault is generated when the Temando API service is unable to process credit card payments. This will normally be a temporary issue.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Service03</faultcode>
      <faultstring>The Temando system is experiencing difficulties in
        processing credit card payments. Please try again
        later.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.4. Err_Connection01

This fault is generated when the Temando API service is being access using an invalid domain name (eg. <https://something.temando.com/soap/>).

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Connection01</faultcode>
```

```
<faultstring>Invalid access for service.</faultstring>
<faultactor>http://www.temando.com/soap</faultactor>
</soap-env:Fault>
</soap-env:Body>
</soap-env:Envelope>
```

2.7.5. Err_Connection02

This fault is generated when the Temando API service is not being accessed securely, ie. https.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Connection02</faultcode>
      <faultstring>This service can only be view securely, ie.
        https.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.6. Err_Connection03

This fault is generated when the Temando API service has not been called via an http post.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Connection03</faultcode>
      <faultstring>Invalid access for service.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.7. Err_Access01

This fault is generated when a Temando API service is being called from an IP address that is not included in the list of safe IP addresses.

Please contact Temando for further details on adding IP address to the safe list.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
```

```
<soap-env:Fault>
  <faultcode>soap-env:Err_Access01</faultcode>
  <faultstring>The IP address that this service is being called from is
    not registered for this service.</faultstring>
  <faultactor>http://www.temando.com/soap</faultactor>
</soap-env:Fault>
</soap-env:Body>
</soap-env:Envelope>
```

2.7.8. Err_Access02

This fault is generated when a Temando API service call does not contain a header or the information containing the Username and Password was either not provided, incomplete or incorrectly formatted.

See section 2.2 User Authentication for further details on sending authentication details via the Header.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access02</faultcode>
      <faultstring>Header information containing the Username and Password
        was either not provided, incomplete or incorrectly
        formatted.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.9. Err_Access03

This fault is generated when the supplied Username and Password are not valid, eg. incorrect password.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access03</faultcode>
      <faultstring>The login details you have provided are not a valid
        Username and Password combination.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.10. Err_Access04

This fault is generated when the supplied Username and Password correspond to an account that has been locked out of the Temando system – due to the maximum number of allowed failed login attempts being reached.

You will need to contact Temando for further details and to unlock your account.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access04</faultcode>
      <faultstring>The login details you have provided are for an account
        that has been locked - due to the maximum number of allowed
        failed login attempts being reached.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.11. Err_Access05

This fault is generated when the supplied Username and Password correspond to an account that has been locked out of the Temando system – due to the account registration awaiting approval by system administrators. This fault will only occur when connecting to the system as a Carrier.

You will need to wait for the processing of your registration request before being able to use this service.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access05</faultcode>
      <faultstring>The login details you have provided are for an account
        that has been locked - due to the account registration awaiting
        approval by system administrators.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.12. Err_Access06

This fault is generated when the supplied Username and Password correspond to an account that has been locked out of the Temando system – due to inappropriate activities.

You will need to contact Temando for further details and to unlock your account.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access06</faultcode>
      <faultstring>The login details you have provided are for an account
        that has been locked - due to inappropriate
        activities.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.13. Err_Access07

This fault is generated when the supplied Username and Password correspond to an account that does not have permission to utilise this service.

You will need to contact Temando to organise the setup of this access right.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access07</faultcode>
      <faultstring>The login details you have provided are for an account
        that is not authorised to use the API.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.14. Err_Access08

This fault is generated when the supplied Username and Password correspond to an account that does not have permission to call a specific operation.

You will need to contact Temando to organise the setup of this access right.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access08</faultcode>
      <faultstring>The login details you have provided are for an account
        that is not authorised to use the API call
        &quot;getQuotesByRequest&quot;.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.15. Err_Access09

This fault is generated when the supplied Username and Password correspond to an account that requires the password to be changed. If this exception is generated the user will required to log into the website and change their password before API facilities will be available to them.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Access09</faultcode>
      <faultstring>The login details you have provided are for an account
        that requires the password to be changed.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.16. Err_Format

This fault is generated when any of the supplied elements contain invalid data.

The specific details of the format issue/s are detailed in the *<faultstring>* element. Some examples of the type of issue strings returned include:

- The 'Username' element (within the 'UsernameToken' type) does not contain a value. This is a mandatory element.
- The 'class' element (within the 'Anything' type) does not contain a valid value. The value must be one of the following: Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other.
- The 'subclass' element (within the 'Anything' type) does not contain a valid value. If the 'class' element contains a 'General Goods' value then this elements value must be one of the following: Household Goods, Excess Baggage, Furniture or Other (Etc.).
- The 'Username' element (within the 'UsernameToken' type) does not contain a valid value. This must be a string between 1 and 64 characters long.
- The 'qualifierVehicleMake', 'qualifierVehicleModel', 'qualifierVehicleDescription' and 'qualifierVehicleYear' elements (within the 'Anything' type) do not contain valid values. These values must match with the predefined settings in the Temando system.
- The 'quantity' element (within the 'Anything' type) does not contain a valid value. This must be a number with no more than 5 digits.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
```

```

<faultcode>soap-env:Err_Format</faultcode>
<faultstring>The 'Username' element (within the
'UsernameToken' type) does not contain a value. This
is a mandatory element.</faultstring>
<faultactor>http://www.temando.com/soap</faultactor>
</soap-env:Fault>
</soap-env:Body>
</soap-env:Envelope>

```

As this is a generic exception that can be applied to multiple entries and combinations each Err_Format exception will not be listed in individual sections. However, to assist in error handling the following table summarises the different types that can be thrown.

In the messages listed each entry that is represented as *[example]* (ie. Italic and within square brackets) will be replaced by the relevant value that relates to the error being generated.

For example if request was supplied and the “class” field did not contain a value and this is a mandatory field then fault code Err_Format01 would be thrown and the message would be “The 'class' element (within the 'anything' type) does not contain a value. This is a mandatory element.”.

Fault Code	Message
Err_Format01	The ' <i>[tag name]</i> ' element (within the ' <i>[parent tag name]</i> ' type) does not contain a value. This is a mandatory element.
Err_Format02	The ' <i>[tag name]</i> ' element (within the ' <i>[parent tag name]</i> ' type) does not contain a valid value. The value must be one of the following: <i>[list of options]</i> .
Err_Format03	The ' <i>[tag name 1]</i> ' element (within the ' <i>[parent tag name]</i> ' type) does not contain a valid value. If the ' <i>[tag name 2]</i> ' element contains a ' <i>[value]</i> ' value then this elements value must be one of the following: <i>[list of options]</i>
Err_Format04	The ' <i>[tag name]</i> ' element (within the ' <i>[parent tag name]</i> ' type) does not contain a valid value. This must be a string between <i>[minimum number of characters]</i> and <i>[maximum number of characters]</i> characters long.
Err_Format05	The ' <i>[tag name 1]</i> ' and ' <i>[tag name 2]</i> ' elements (within the ' <i>[parent tag name]</i> ' type) do not contain valid values. These values must match with the predefined settings in the Temando system.
Err_Format06	The ' <i>[tag name 2]</i> ', ' <i>[tag name 2]</i> ' and ' <i>[tag name 3]</i> ' elements (within the ' <i>[parent tag name]</i> ' type) do not contain valid values. These values must match with the predefined settings in the Temando system.
Err_Format07	The ' <i>[tag name 3]</i> ', ' <i>[tag name 2]</i> ', ' <i>[tag name 3]</i> ' and ' <i>[tag name 4]</i> ' elements (within the ' <i>[parent tag name]</i> ' type) do not contain valid values. These values must match with the predefined settings in the Temando system.
Err_Format08	The ' <i>[tag name]</i> ' element (within the ' <i>[parent tag name]</i> ' type) does not contain a valid value. This must be a number with no more than <i>[maximum number of digits]</i> digits.

Err_Format09	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. Dates must be of the format yyyy-mm-dd (eg. for 21st February, 2010 the format would be 2010-02-21).
Err_Format10	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The date provided does not represent a real date.
Err_Format11	The '[tag name 1]' and '[tag name 2]' elements (within the '[parent tag name]' type) contain invalid values. The datetime represented is a date and time before the current date and time.
Err_Format12	The '[tag name 1]' and '[tag name 2]' elements (within the '[parent tag name]' type) contain invalid values. The datetime represented is a date and time more than 5 years in the future. Send dates must not exceed this maximum.
Err_Format13	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. Dates must be of the format mm-yyyy (eg. for expiry February, 2010 the format would be 02-2010).
Err_Format14	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The date provided has expired.
Err_Format15	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The card number provided is not a valid number based on the type of card provided.
Err_Format16	The '[tag name]' location you have supplied does not match with the '[tag name 2]' details provided (ie. Anywhere) to generate the quote price selected.
Err_Format17	The '[tag name 1]' and '[tag name 2]' elements (within the '[parent tag name]' type) contain invalid values. The date and time requested is being lodged after the cutoff datetime of '[cutoff datetime]' for the specified origin.
Err_Format18	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The '[tag name]' supplied is already being used by another account and cannot be used.
Err_Format19	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The date supplied represents a date of birth that is after the current date.
Err_Format20	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The email/s supplied do not represent valid email addresses (Note: multiple email addresses must be separated by a ',' character).
Err_Format21	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. The supplied '[tag name]' is already associated with another account and cannot be used.
Err_Format22	The '[tag name]' element (within the '[parent tag name]' type) contain invalid values. The date represented is a date after the current date.
Err_Format23	The '[tag name]' element (within the '[parent tag name]' type) does not contain a valid value. This must be a number with no more than '[maximum number of digits]' digits and at most '[maximum number of decimals]' decimal values.

Err_Format24	The '[tag name]' element (within the '[parent tag name]' type) contains an invalid value. Datetimes must be of the format yyyy-mm-ddTHH:ii:SS (eg. for UTC 7:23pm (and 44seconds) on the 21st February, 2010 the format would be 2010-02-21T19:23:44Z).
Err_Format25	The '[tag name 1]' element for insurance has been included for the quote however a value has not been supplied for the '[tag name 2]' (within the '[parent tag name]' type). A value must be supplied for the '[tag name 2]' if insurance is to be included as an '[tag name 1]'.
Err_Format26	The '[tag name 1]' element (within the '[parent tag name 1]' type) contains a post office box address. As a result the '[tag name 2]' element (within the '[parent tag name 2]' type) must be supplied and contain the value 'Y'. Note: this change may result in a 'Quote does not match ...' error (ie. Err_State05) because the initial request for quotes may not have specified that the destination was a post office box address and the carrier selected for booking does not support post office box addresses.
Err_Format27	The '[tag name 1]' element (within the '[parent tag name 1]' type) contains a description for a location that already exists.
Err_Format28	The '[tag name 1]' element (within the '[parent tag name 1]' type) contains a description for a location that does not exist.

2.7.17. Err_Payment01

This fault is generated when the payment method nominated by the user is credit and there is insufficient funds remaining in available credit to process this request.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Payment01</faultcode>
      <faultstring>The amount for the quote exceeds the amount of credit
        remaining. The remaining credit is AUD$23.14.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.18. Err_Payment02

This fault is generated when the payment method nominated by the user is account and they do not have account facilities enabled.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap-env:Body>
  <soap-env:Fault>
    <faultcode>soap-env:Err_Payment02</faultcode>
    <faultstring>You do not have an account available with Temando.
      Please contact Temando for further details.</faultstring>
    <faultactor>http://www.temando.com/soap</faultactor>
  </soap-env:Fault>
</soap-env:Body>
</soap-env:Envelope>
```

2.7.19. Err_Payment03

This fault is generated when the payment method nominated by the user is account and the amount of the request will result in the total amount for transactions (for the account period) exceeding their allowed maximum.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Payment03</faultcode>
      <faultstring>The amount for the quote exceeds the amount of available
        funds left in your account. The remaining funds are
        AUD$23.14.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.20. Err_Payment04

This fault is generated when the payment method nominated by the user is credit card and either the details were invalid or there is insufficient funds.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Payment04</faultcode>
      <faultstring>The credit card payment has been declined. This could
        Be due to invalid details or insufficient funds.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.21. Err_Permission01

This fault is generated when a specific address (either the origin or destination) has been flagged by Temando and items are prevented from being sent to and received from this location.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Permission01</faultcode>
      <faultstring>The Origin location you have supplied has been flagged
        by Temando and items are prevented from being sent to and
        received from this location. If you require further information
        on this please contact Temando.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.22. Err_Permission02

This fault is generated when a user attempts to access information relating to a request that is not related to their account.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Permission02</faultcode>
      <faultstring>You do not have permission to access the request
        associated with the supplied reference.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.23. Err_State01

This fault is generated when a user attempts to access information relating to a request that does not exist or is incomplete.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State01</faultcode>
      <faultstring>A request with the supplied reference does not exist or
        does not contain all the necessary details.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.24. Err_State02

This fault is generated when a user attempts to access information relating to a request that is not accessible via the Temando API service. An example of when this may occur is if a request is created via the website and then attempting to access this request via an API operation.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State02</faultcode>
      <faultstring>The request associated with the supplied reference was
        not generated via the API and can not be processed via this API
        access right.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.25. Err_State03

This fault is generated when a user attempts to access information relating to a request or perform some other action on the request when the request does not have the correct status for the operation being called from the Temando API service. An example of when this may occur is if a request has been completed (ie. paid) and the user has attempted to make a payment against this request.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State03</faultcode>
      <faultstring>The request associated with the supplied reference does
        not have the correct status for the call requesting it. The
        current status of this request is [current status]. For the
        call used the request must have a status of
        [required status].</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.26. Err_State04

This fault is generated when a user attempts to use a promotion code that:

- does not exist; or
- is not available via this service (ie. API); or
- has expired or is no longer available.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State04</faultcode>
      <faultstring>The promotion code provided does not exist, is not
        available via this service or has expired and is no longer
        available.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.27. Err_State05

Thrown from Operations:

- makeBookingByRequest

This fault is generated when a user attempts to book a quote and the details of the quote to book do not match with any available quotes within the Temando system. The 2 main reasons why this could occur are:

1. The supplied quote is no longer available (ie. the Carrier has changed their pricing details since the getQuotesByRequest was made); or
2. The details of the supplied quote do not match exactly (ie. is case-sensitive) with the quote supplied from the getQuotesByRequest call.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State05</faultcode>
      <faultstring>The quote selected for the booking does not match with
        an available quote based on the provided details (ie. Anything,
        Anywhere and Anytime). If you have selected the quote from the
        getQuotesByRequest call it is possible that the quote is no longer
        available. If not, please check that the details of the booking
        quote match EXACTLY (ie. case-sensitive) with those of the
        corresponding available quote selected.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.28. Err_State06

This error will no longer be thrown.

Thrown from Operations:

- makeBookingById

This fault is generated when a user attempts to book a quote (without supplying a quote id) and the associated request does not have an accepted quote. The main reason why this could occur is because the user created a request via the plugin but did not accept an available quote or no quotes were available.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State06</faultcode>
      <faultstring>An accepted quote associated with the supplied request
        id does not exist or does not contain all the necessary
        details.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.29. Err_State07

Thrown from Operations:

- addBookingDetails

This fault is generated when a user attempts to assign a booking reference number to a request that is not in the correct state (eg. payment has not yet been received) or already has a booking number.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State07</faultcode>
      <faultstring> The request associated with the supplied id does not
        have the correct status. This may be due to a payment not yet
        being made for this request or this request already having a
        booking reference.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.30. Err_State08

This error will no longer be thrown.

Thrown from Operations:

- makeBookingById

This fault is generated when a user attempts to book a quote for a request when the quote id supplied does not match with the accepted quote. The main reason why this could occur is because the user created a request via the plugin, accepted a certain quote and then an attempt is made to accept another quote.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State08</faultcode>
      <faultstring>An accepted quote associated with the supplied reference
        does not exist or does not contain all the necessary
        details.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.31. Err_State09

Thrown from Operations:

- addBookingDetails

This fault is generated when a user attempts to assign a consignment number to a request and this consignment number has already been assigned to another request.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State09</faultcode>
      <faultstring>The consignment number supplied has already been assigned
        to another request.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.32. Err_State10

Thrown from Operations:

- addBookingDetails

This fault is generated when a user attempts to assign a booking reference number to a request and this booking reference number has already been assigned to another request.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State10</faultcode>
      <faultstring>The booking reference number supplied has already been
        assigned to another request.</faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.33. Err_State11

Thrown from Operations:

- makeBookingByRequest

This fault is generated when a user attempts to book a quote and the details of an extra to be included does not match with any available extras within the Temando system. The 2 main reasons why this could occur are:

1. The supplied extra is no longer available (ie. the insurance conditions have changed since the getQuotesByRequest was made); or
2. The details of the supplied extra do not match exactly (ie. is case-sensitive) with the quote supplied from the getQuotesByRequest call.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State11</faultcode>
      <faultstring>The extra supplied for the booking does not match with
        an available extra based on the provided details (ie. Anything,
        Anywhere, Anytime and General). If you have selected the extra
        from the getQuotesByRequest call it is possible that the extra
        is no longer available. If not, please check that the details
        of the booking extra match EXACTLY (ie. case-sensitive) with
        those of the corresponding available quote selected.
      </faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.1. Err_State12

Thrown from Operations:

- getRequest
- cancelRequest

This fault is generated when a user attempts to access information relating to a request using a client specified reference and the reference is associated with more than one request (ie. is not unique).

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_State12</faultcode>
      <faultstring>More than one request with the supplied reference exists.
        For this call this identifier must be unique.
      </faultstring>
      <faultactor>http://www.temando.com/soap</faultactor>
    </soap-env:Fault>
  </soap-env:Body>
</soap-env:Envelope>
```

2.7.2. Err_Data01

Thrown from Operations:

- getQuotesByRequest
- makeBookingByRequest

This fault is generated when a user attempts to supply details that match with an unavailable date and/or location for pickup or drop off at a depot. This may occur if services are not provided on specific days or dates for a given location.

Examples:

- General freight services are not available on Saturdays or Sundays.
- There is a general public holiday on the 01/01/2010 in Australia for New Year's Day and no services are available on this date.
- There is a general public holiday on the 14/06/2010 in Queensland, Australia for the Queen's Birthday and no services are available on this date for this state. All other states (eg. NSW, Victoria, etc) are not affected.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>soap-env:Err_Data01</faultcode>
      <faultstring>Unable to send on the specified date and/or at the
```

```
location because: New Years Day (Australian public  
holiday).</faultstring>  
<faultactor>http://www.temando.com/soap</faultactor>  
</soap-env:Fault>  
</soap-env:Body>  
</soap-env:Envelope>
```

2.8. Common Types

This section describes all the common schema types used by the API operations. The normative definitions for these types are found in the common.xsd xml schema (http://api.temando.com/schema/2009_06/common.xsd), the relevant parts of which are excerpted here.

2.8.1. Compound Data Types

2.8.1.1. Request

```
<xsd:complexType name="Request">
  <xsd:all>
    <xsd:element name="status" type="com:RequestStatus"/>
    <xsd:element name="anythings">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="anything" type="com:Anything" maxOccurs="10"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="anywhere" type="com:Anywhere"/>
    <xsd:element name="anytime" type="com:Anytime"/>
    <xsd:element name="general" type="com:General" minOccurs="0"/>
    <xsd:element name="origin" type="com:Location"/>
    <xsd:element name="destination" type="com:Location"/>
    <xsd:element name="quotes">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="quote" type="com:AvailableQuote" minOccurs="0"
            maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="instructions" type="com:Instructions"
      minOccurs="0"/>
    <xsd:element name="reference" type="com:ClientReference"
      minOccurs="0"/>
    <xsd:element name="comments" type="com:Comments" minOccurs="0"/>
  </xsd:all>
  <xsd:attribute name="id" type="xsd:positiveInteger"/>
</xsd:complexType>
```

Child content:

The specific details about an item/s being sent.

Field	Base type	Occurs	Description
id (<i>attribute</i>)	positiveInteger	Optional	Attribute: Unique identifier for the request.
status	simple	Always	The status of the request. See section 2.8.2.85 RequestStatus for further details.
anythings	list	Always	The list of anything details of the item/s being sent. See section 2.8.1.2 Anything for further details.

anywhere	complex	Always	The anywhere details of the item/s being sent. See section 2.8.1.3 Anywhere for further details.
anytime	complex	Always	The anytime details of the item/s being sent. See section 2.8.1.4 Anytime for further details.
general	complex	Optional	The general details of the item/s being sent. See section 2.8.1.5 General for further details.
origin	complex	Always	The origin details of the item/s being sent. See section 2.8.1.12 Location for further details.
destination	complex	Always	The destination details of the item/s being sent. See section 2.8.1.12 Location for further details.
quotes	list	Always	The list of quotes for the item/s being sent. See section 2.8.1.6 AvailableQuote for further details.
instructions	simple	Optional	The delivery instructions relating to the item/s being sent. See section 2.8.2.33 Instructions for further details.
reference	simple	Optional	The user supplied reference relating to the request. See section 2.8.2.102 ClientReference for further details.
comments	simple	Optional	The user supplied comments relating to the item/s being sent. See section 2.8.2.34 Comments for further details.

Used in: 2.6.3 getRequest, 2.6.4 getRequestsRequiringBooking and 2.6.10 getManifest.

Example:

```
<request id="100000" type="com:Request">
  <status>Completed</status>
  <anythings>
    <anything>
      <class>Vehicle</class>
      <subclass>Car</subclass>
      <qualifierVehicleMake>Ford</qualifierVehicleMake>
      <qualifierVehicleModel>Mustang</qualifierVehicleModel>
      <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
        289</qualifierVehicleDescription>
      <qualifierVehicleRunning>Y</qualifierVehicleRunning>
      <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
      <qualifierVehicleYear>1965</qualifierVehicleYear>
      <quantity>1</quantity>
    </anything>
  </anythings>
  <anywhere>
    <itemNature>Domestic</itemNature>
    <itemMethod>Door to Door</itemMethod>
    <originCountry>AU</originCountry>
    <originCode>4000</originCode>
```

```

<originSuburb>Brisbane</originSuburb>
<destinationCountry>AU</destinationCountry>
<destinationCode>2000</destinationCode>
<destinationSuburb>Sydney</destinationSuburb>
<destinationIs>Residence</destinationIs>
<destinationResPostalBox>N</destinationResPostalBox>
<destinationResUnattended>N</destinationResUnattended>
<destinationResInside>N</destinationResInside>
<destinationResLimitedAccess>N</destinationResLimitedAccess>
<destinationResHeavyLift>N</destinationResHeavyLift>
<originIs>Business</originIs>
<originBusUnattended>N</originBusUnattended>
<originBusDock>Y</originBusDock>
<originBusForklift>N</originBusForklift>
<originBusLoadingFacilities>N</originBusLoadingFacilities>
<originBusInside>N</originBusInside>
<originBusLimitedAccess>N</originBusLimitedAccess>
<originBusHeavyLift>N</originBusHeavyLift>
<originBusTailgateLifter>N</originBusTailgateLifter>
<originBusContainerSwingLifter>N</originBusContainerSwingLifter>
</anywhere>
<anytime>
<readyDate>2009-07-01</readyDate>
<readyTime>PM</readyTime>
</anytime>
<general>
<goodsValue>2000.00</goodsValue>
</general>
<origin>
<description>ABC Suppliers</description>
<contactName>John Doe</contactName>
<companyName>ABC Suppliers</companyName>
<street>1 Originating Street</street>
<suburb>Brisbane</suburb>
<state>QLD</state>
<code>4000</code>
<country>AU</country>
<phone1>(07) 3333 3333</phone1>
<phone2></phone2>
<fax>(07) 3333 3331</fax>
<email>bneoffice@abcsuppliers.com.au</email>
</origin>
<destination>
<contactName>Jane Doe</contactName>
<street>2 Recipient Street</street>
<suburb>Sydney</suburb>
<state>NSW</state>
<code>2000</code>
<country>AU</country>
<phone1>(02) 9333 3333</phone1>
<phone2></phone2>
<fax>(02) 9333 3331</fax>
<email>jane.doe@yahoo.com.au</email>
</destination>
<quotes>
<quote id="111111">
<generated>Automatically</generated>
<accepted>Y</accepted>
<bookingNumber>XYZ300001</bookingNumber>
<consignmentNumber>UVW400001</consignmentNumber>
<consignmentDocument>JVBER ... shortened for readability ... XRhbG
</consignmentDocument>
<consignmentDocumentType>application/pdf</consignmentDocumentType>
<labelDocument>e3Rf3 ... shortened for readability ... X8ggy
</labelDocument>
<labelDocumentType>application/pdf</labelDocumentType>
<trackingStatus>In Transit</trackingStatus>
<trackingStatusOccurred>2009-07-02T15:25:46Z
</trackingStatusOccurred>

```

```

<trackingLastChecked>2009-07-02T18:13:12Z
  </trackingLastChecked>
<trackingFurtherDetails></trackingFurtherDetails>
<totalPrice>110.00</totalPrice>
<basePrice>100.00</basePrice>
<tax>10.00</tax>
<currency>AUD</currency>
<deliveryMethod>Off-peak</deliveryMethod>
<usingGeneralRail>N</usingGeneralRail>
<usingGeneralRoad>Y</usingGeneralRoad>
<usingGeneralSea>N</usingGeneralSea>
<usingExpressAir>N</usingExpressAir>
<usingExpressRoad>N</usingExpressRoad>
<etaFrom>2</etaFrom>
<etaTo>3</etaTo>
<guaranteedEta>Y</guaranteedEta>
<carrierId>654321</carrierId>
<extras>
  <extra>
    <summary>Insurance</summary>
    <totalPrice>22.00</totalPrice>
    <basePrice>20.00</basePrice>
    <tax>2.00</tax>
  </extra>
</extras>
</quote>
</quotes>
<instructions>Please use side door.</instructions>
<reference>ABC200000</reference>
<comments>Please don't use any chains on the axle.</comments>
</request>

```

2.8.1.2. Anything

```

<xsd:complexType name="Anything">
  <xsd:all>
    <xsd:element name="class" type="tns:Class"/>
    <xsd:element name="subclass" type="tns:Subclass" minOccurs="0"/>
    <xsd:element name="mode" type="tns:Mode" minOccurs="0"/>
    <xsd:element name="tlSubclass" type="tns:TlSubclass" minOccurs="0"/>
    <xsd:element name="packaging" type="tns:Packaging" minOccurs="0"/>
    <xsd:element name="palletType" type="tns:PalletType" minOccurs="0"/>
    <xsd:element name="palletNature" type="tns:PalletNature" minOccurs="0"/>
    <xsd:element name="containerDimensions" type="tns:ContainerDimensions"
      minOccurs="0"/>
    <xsd:element name="containerNature" type="tns:ContainerNature"
      minOccurs="0"/>
    <xsd:element name="containerRegistered" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="qualifierFreightGeneralDangerousGoods"
      type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="qualifierFreightGeneralFragile"
      type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="qualifierFreightGeneralRefrigerated"
      type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="qualifierVehicleMake" type="tns:VehicleMake"
      minOccurs="0"/>
    <xsd:element name="qualifierVehicleModel" type="tns:VehicleModel"
      minOccurs="0"/>
    <xsd:element name="qualifierVehicleDescription"
      type="tns:VehicleDescription" minOccurs="0"/>
    <xsd:element name="qualifierVehicleRunning" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="qualifierVehicleRegistered" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="qualifierVehicleYear" type="tns:VehicleYear"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>

```

```

<xsd:element name="qualifierVehicleRegistration"
  type="tns:VehicleRegistration" minOccurs="0"/>
<xsd:element name="qualifierBoatMake" type="tns:BoatMake"
  minOccurs="0"/>
<xsd:element name="qualifierBoatModel" type="tns:BoatModel"
  minOccurs="0"/>
<xsd:element name="qualifierBoatSeaworthy" type="tns:YesNoOption"
  minOccurs="0"/>
<xsd:element name="qualifierBoatTrailer" type="tns:YesNoOption"
  minOccurs="0"/>
<xsd:element name="qualifierBoatHullType" type="tns:BoatHullType"
  minOccurs="0"/>
<xsd:element name="qualifierLivestockType" type="tns:LivestockType"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalVaccinated" type="tns:YesNoOption"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalRegistered" type="tns:YesNoOption"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalType" type="tns:AnimalType"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalBreed" type="tns:AnimalBreed"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalSex" type="tns:Sex" minOccurs="0"/>
<xsd:element name="qualifierAnimalAge" type="tns:AnimalAge"
  minOccurs="0"/>
<xsd:element name="qualifierAnimalCrate" type="tns:AnimalCrate"
  minOccurs="0"/>
<xsd:element name="distanceMeasurementType"
  type="tns:DistanceMeasurementType" minOccurs="0"/>
<xsd:element name="weightMeasurementType"
  type="tns:WeightMeasurementType" minOccurs="0"/>
<xsd:element name="length" type="tns:Length" minOccurs="0"/>
<xsd:element name="width" type="tns:Width" minOccurs="0"/>
<xsd:element name="height" type="tns:Height" minOccurs="0"/>
<xsd:element name="weight" type="tns:Weight" minOccurs="0"/>
<xsd:element name="quantity" type="tns:Quantity"/>
<xsd:element name="description" type="tns:ItemDescription"
  minOccurs="0"/>
</xsd:all>
</xsd:complexType>

```

Child content:

Specific details about an item being sent.

Field	Base type	Occurs	Description
class	simple	Always	The class of the item being sent. See section 2.8.2.3 Class for further details. Note: current implementation only supports: Freight, General Goods, Vehicle and Other (does not support: Boat, Animal and Refrigerated).
subclass	simple	Optional [*]	The subclass of the item being sent. See section 2.8.2.4 Subclass for further details.
mode	simple	Optional [*]	The mode of the item being sent. See section 2.8.2.5 Mode for further details.
tlSubclass	simple	Optional [*]	The number of pallets being sent. See section 2.8.2.6 TISubclass for further details.
packaging	simple	Optional [*]	The type of packaging used for the item being sent.

			See section 2.8.2.7 Packaging for further details.
palletType	simple	Optional [*]	The pallet type being sent. See section 2.8.2.8 PalletType for further details.
palletNature	simple	Optional [*]	The pallet nature being sent. See section 2.8.2.9 PalletNature for further details.
containerDimensions	simple	Optional [*]	The dimensions of the container being sent. See section 2.8.2.10 ContainerDimensions for further details.
containerNature	simple	Optional [*]	The nature of the container being sent. See section 2.8.2.11 ContainerNature for further details.
containerRegistered	simple	Optional [*]	Whether the container is registered. See section 2.8.2.1 YesNoOption for further details.
qualifierFreightGeneralFragile	simple	Optional [*]	If the item being sent is fragile. See section 2.8.2.1 YesNoOption for further details.
qualifierVehicleMake	simple	Optional [*]	The make of the vehicle being sent. See section 2.8.2.12 VehicleMake for further details.
qualifierVehicleModel	simple	Optional [*]	The model of the vehicle being sent. See section 2.8.2.13 VehicleModel for further details.
qualifierVehicleDescription	simple	Optional [*]	The description of the vehicle being sent. See section 2.8.2.14 VehicleDescription for further details.
qualifierVehicleYear	simple	Optional [*]	The year of the vehicle being sent. See section 2.8.2.15 VehicleYear for further details.
qualifierVehicleRunning	simple	Optional [*]	If the vehicle being sent is running. See section 2.8.2.1 YesNoOption for further details.
qualifierVehicleRegistered	simple	Optional [*]	If the vehicle being sent is registered. See section 2.8.2.1 YesNoOption for further details.
qualifierVehicleRegistration	simple	Optional [*]	The registration of the vehicle being sent. See section 2.8.2.16 VehicleRegistration for further details.
qualifierBoatSeaworthy	simple	Optional [*]	If the boat being sent if seaworthy. See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support the class Boat so any use of this element will be ignored.
qualifierBoatTrailer	simple	Optional [*]	If the boat being sent has a trailer. See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support the class Boat so any use of this element will be ignored.
qualifierBoatMake	simple	Optional [*]	The make of the boat being sent. See section 2.8.2.17 BoatMake for further details. Note: current implementation does not support the class Boat so any use of this element will be ignored.

qualifierBoatModel	simple	Optional [*]	The model of the boat being sent. See section 2.8.2.18 BoatModel for further details. Note: current implementation does not support the class Boat so any use of this element will be ignored.
qualifierBoatHullType	simple	Optional [*]	The hull type of the boat being sent. See section 2.8.2.19 BoatHullType for further details. Note: current implementation does not support the class Boat so any use of this element will be ignored.
qualifierLivestockType	simple	Optional [*]	The type of the livestock being sent. See section 2.8.2.20 LivestockType for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalVaccinated	simple	Optional [*]	If the animal being sent has been vaccinated. See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalAge	simple	Optional [*]	The age of the animal being sent. See section 2.8.2.23 AnimalAge for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalRegistered	simple	Optional [*]	If the animal being sent is registered. See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalBreed	simple	Optional [*]	The breed of the animal being sent. See section 2.8.2.22 AnimalBreed for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalType	simple	Optional [*]	The type of the animal being sent. See section 2.8.2.21 AnimalType for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalSex	simple	Optional [*]	The sex of the animal being sent. See section 2.8.2.2 Sex for further details. Note: current implementation does not support the class Animal so any use of this element will be ignored.
qualifierAnimalCrate	simple	Optional [*]	The type of crate to be used to transport the animal being sent. See section 2.8.2.24 AnimalCrate for further details. Note: current implementation does not support the class Animal so any use of

			<i>this element will be ignored.</i>
length	simple	Optional [*]	The length of the item being sent. See section 2.8.2.27 Length for further details.
width	simple	Optional [*]	The width of the item being sent. See section 2.8.2.28 Width for further details.
height	simple	Optional [*]	The height of the item being sent. See section 2.8.2.29 Height for further details.
weight	simple	Optional [*]	The weight of the item being sent. See section 2.8.2.30 Weight for further details.
distanceMeasurementType	simple	Optional [*]	The distance measurement type to be used for the Length, Width and Height values. See section 2.8.2.25 DistanceMeasurementType for further details.
weightMeasurementType	simple	Optional [*]	The weight measurement type to be used for the Weight values. See section 2.8.2.26 WeightMeasurementType for further details.
quantity	simple	Always	The number of items being sent. See section 2.8.2.31 Quantity for further details.
description	simple	Optional [*]	Further details of the item being sent, that have not been specified in other elements. See section 2.8.2.32 ItemDescription for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further Restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
subclass	<ol style="list-style-type: none"> Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is: General Goods, Vehicle, Boat, Animal, Refrigerated or Other. Values will be limited in the following cases: <ul style="list-style-type: none"> If class is "General Goods" options are limited to: Household Goods, Excess Baggage, Furniture, or Other (Etc.); or If class is "Vehicle" options are limited to: Car, Motorcycle, Tractor, Equipment, or Other (Etc.).
mode	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Freight.
tlSubclass	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Freight.
packaging	<ol style="list-style-type: none"> Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is General Goods or Other; or class is Freight and mode is Less than load. Values will be limited in the following cases: <ul style="list-style-type: none"> If class is "General Goods" and mode is "Household Goods" options are

	<p>limited to: Box, Carton, Crate, Cylinder, Document Envelope, Flat Pack, Letter, Pallet, Parcel, Satchel/Bag, Skid, Unpackaged or N/A, or Wheel/Tyre; or</p> <ul style="list-style-type: none"> • If class is "General Goods" and mode is "Excess Baggage" options are limited to: Backpack, Box, Carton or Suitcase; or • If class is "General Goods" and mode is "Excess Furniture" options are limited to: Box, Carton, Crate, Flat Pack, Pallet, Skid or Unpackaged or N/A; or • If class is "General Goods" and mode is "Other (Etc.)" options are limited to: Box, Carton, Crate, Cylinder, Document Envelope, Flat Pack, Letter, Pallet, Parcel, Pipe, Satchel/Bag, Skid, Tube, Unpackaged or N/A, or Wheel/Tyre; or • If class is "Other" options are limited to: Bale, Box, Bunch, Bundle, Carton, Crate, Cylinder, Document Envelope, Drum, Flat Pack, Letter, Pail, Pallet, Parcel, Pipe, Roll, Satchel/Bag, Skid, Suitcase, Tube, Unpackaged or N/A, or Wheel/Tyre; or • If class is "Freight" and mode is "Less than load" options are limited to: Bale, Box, Bunch, Bundle, Carton, Crate, Cylinder, Document Envelope, Drum, Flat Pack, Letter, Pail, Pallet, Parcel, Pipe, Roll, Satchel/Bag, Skid, Suitcase, Tube, or Wheel/Tyre.
palletType	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • class is Refrigerated; or • packaging is Pallet; or • mode is Truckload.
palletNature	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • class is Refrigerated; or • packaging is Pallet; or • mode is Truckload.
containerDimensions	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • mode is Container.
containerNature	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • mode is Container.
containerRegistered	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • mode is Container.
qualifierFreightGeneralFragile	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • class is: Freight, General Goods or Refrigerated.
qualifierVehicleMake	<p>1. Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • class is Vehicle. <p>2. Values will be limited in the following cases:</p> <ul style="list-style-type: none"> • If subclass is "Car" must match exactly (ie. letter for letter but not case sensitive) with a Make defined within the Temando system, which is extracted from The Red Book manual as provided by The Red Book© (https://www.redbook.com.au).
qualifierVehicleModel	<p>1. Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> • class is Vehicle. <p>2. Values will be limited in the following cases:</p> <ul style="list-style-type: none"> • If subclass is "Car" must match exactly (ie. letter for letter but not case sensitive) with a Model defined within the Temando system, which is extracted from The Red Book manual as provided by The Red Book© (https://www.redbook.com.au).
qualifierVehicleDescription	<p>1. Applies only if:</p> <ul style="list-style-type: none"> • class is Vehicle. <p>2. Values will be limited in the following cases:</p> <ul style="list-style-type: none"> • If subclass is "Car" must match exactly (ie. letter for letter but not case sensitive) with a Description defined within the Temando system, which is extracted from The Red Book manual as provided by The Red Book© (https://www.redbook.com.au).
qualifierVehicleYear	<p>1. Applies only if:</p> <ul style="list-style-type: none"> • class is Vehicle.

	<p>2. Values will be limited in the following cases:</p> <ul style="list-style-type: none"> If subclass is "Car" must match exactly (ie. letter for letter but not case sensitive) with a Year defined within the Temando system, which is extracted from The Red Book manual as provided by The Red Book© (https://www.redbook.com.au).
qualifierVehicleRunning	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Vehicle.
qualifierVehicleRegistered	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Vehicle.
qualifierVehicleRegistration	Applies only if: <ul style="list-style-type: none"> class is Vehicle.
qualifierBoatSeaworthy	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Boat.
qualifierBoatTrailer	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Boat.
qualifierBoatMake	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Boat.
qualifierBoatModel	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Boat.
qualifierBoatHullType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Boat.
qualifierLivestockType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal and subclass is Livestock.
qualifierAnimalVaccinated	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal.
qualifierAnimalAge	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal.
qualifierAnimalRegistered	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal.
qualifierAnimalBreed	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal.
qualifierAnimalType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal and subclass is Other.
qualifierAnimalSex	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal.
qualifierAnimalCrate	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Animal and subclass is Dog, Cat, Bird, Livestock or Other.
length	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is General Goods, Boat, Refrigerated or Other; or class is Freight and mode is Less than load; or class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
width	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is General Goods, Boat, Refrigerated or Other; or class is Freight and mode is Less than load; or class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
height	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is General Goods, Boat, Refrigerated or Other; or class is Freight and mode is Less than load; or class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
weight	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Freight, General Goods, Animal, Boat, Refrigerated or Other; or class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
distanceMeasurementType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is General Goods, Boat, Refrigerated or Other; or class is Freight and mode is Less than load; or class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
weightMeasurementType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Freight, General Goods, Animal, Boat, Refrigerated or Other; or

	<ul style="list-style-type: none"> class is Vehicle and subclass is Tractor, Equipment or Other (Etc.).
description	Applies (and is mandatory) only if: <ul style="list-style-type: none"> class is Freight, General Goods or Refrigerated.

Used in: 2.6.1 getQuotesByRequest and 2.6.2 makeBookingByRequest.

Example:

```
<anything xsi:type="com:Anything">
  <class>Vehicle</class>
  <subclass>Car</subclass>
  <qualifierVehicleMake>Ford</qualifierVehicleMake>
  <qualifierVehicleModel>Mustang</qualifierVehicleModel>
  <qualifierVehicleDescription>GT Convertible 2dr Auto 3sp
    289</qualifierVehicleDescription>
  <qualifierVehicleRunning>Y</qualifierVehicleRunning>
  <qualifierVehicleRegistered>N</qualifierVehicleRegistered>
  <qualifierVehicleYear>1965</qualifierVehicleYear>
  <quantity>1</quantity>
</anything>
```

2.8.1.3. Anywhere

```
<xsd:complexType name="Anywhere">
  <xsd:all>
    <xsd:element name="itemNature" type="tns:DeliveryNature"/>
    <xsd:element name="itemMethod" type="tns:DeliveryType"/>
    <xsd:element name="originCountry" type="tns:CountryCode"/>
    <xsd:element name="originCode" type="tns:PostalCode" minOccurs="0"/>
    <xsd:element name="originSuburb" type="tns:Suburb" minOccurs="0"/>
    <xsd:element name="originState" type="tns:State" minOccurs="0"/>
    <xsd:element name="originCity" type="tns:City" minOccurs="0"/>
    <xsd:element name="originPort" type="tns:PortName" minOccurs="0"/>
    <xsd:element name="destinationCountry" type="tns:CountryCode"/>
    <xsd:element name="destinationCode" type="tns:PostalCode"
      minOccurs="0"/>
    <xsd:element name="destinationSuburb" type="tns:Suburb"
      minOccurs="0"/>
    <xsd:element name="destinationState" type="tns:State" minOccurs="0"/>
    <xsd:element name="destinationCity" type="tns:City" minOccurs="0"/>
    <xsd:element name="destinationPort" type="tns:PortName"
      minOccurs="0"/>
    <xsd:element name="portType" type="tns:PortType" minOccurs="0"/>
    <xsd:element name="destinationIs" type="tns:LocationType"/>
    <xsd:element name="destinationBusDock" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusForklift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusLoadingFacilities"
      type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="destinationBusPostalBox" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusUnattended" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusInside" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusNotifyBefore" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusLimitedAccess" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusHeavyLift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusTailgateLifter" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationBusContainerSwingLifter" type="tns:YesNoOption"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

```

    type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="destinationResPostalBox" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResUnattended" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResInside" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResNotifyBefore" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResLimitedAccess" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResHeavyLift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="destinationResTailgateLifter" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originIs" type="tns:LocationType"/>
    <xsd:element name="originBusUnattended" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusDock" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusForklift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusLoadingFacilities" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusInside" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusNotifyBefore" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusLimitedAccess" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusHeavyLift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusTailgateLifter" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originBusContainersSwingLifter" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResUnattended" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResInside" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResNotifyBefore" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResLimitedAccess" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResHeavyLift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="originResTailgateLifter" type="tns:YesNoOption"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>

```

Child content:

Specific details about where an item/s being sent.

Field	Base type	Occurs	Description
itemNature	simple	Always	<p>Whether the item is being sent domestically (ie. within the same country) or internationally (from one country to another). See section 2.8.2.35 DeliveryNature for further details.</p> <p>Note: current implementation only supports: Domestic (does not support: International).</p>

itemMethod	simple	Always	The delivery method for the item being sent. See section 2.8.2.36 DeliveryType for further details. Note: current implementation only supports: Door to Door and Depot to Depot (does not support: Port to Port).
originCountry	simple	Always	The country that the item is being sent from (international) or within (domestic). See section 2.8.2.42 CountryCode for further details. Note: current implementation only supports: AU.
originCode	simple	Optional*	The postal code of the location that the item is being sent from. See section 2.8.2.41 PostalCode for further details.
originSuburb	simple	Optional*	The suburb of the location that the item is being sent from. See section 2.8.2.38 Suburb for further details.
originState	simple	Optional*	The state of the location that the item is being sent from. See section 2.8.2.39 State for further details.
originCity	simple	Optional*	The city or town of the location that the item is being sent from. See section 2.8.2.40 City for further details.
originPort	simple	Optional*	The port name of the location that the item is being sent from. See section 2.8.2.47 PortName for further details. Note: current implementation does not support the itemNature International so any use of this element will be ignored.
destinationCountry	simple	Always	The country that the item is being sent to (international) or within (domestic). See section 2.8.2.42 CountryCode for further details. Note: current implementation only supports: AU.
destinationCode	simple	Optional*	The postal code of the location that the item is being sent to. See section 2.8.2.41 PostalCode for further details.
destinationSuburb	simple	Optional*	The suburb of the location that the item is being sent to. See section 2.8.2.38 Suburb for further details.
destinationState	simple	Optional*	The state of the location that the item is being sent to. See section 2.8.2.39 State for further details.
destinationCity	simple	Optional*	The city or town of the location that the item is being sent to.

			See section 2.8.2.40 City for further details.
destinationPort	simple	Optional*	The port name of the location that the item is being sent to. See section 2.8.2.47 PortName for further details. Note: current implementation does not support the itemNature International so any use of this element will be ignored.
portType	simple	Optional*	The type of port of the locations that the item is being sent from/to. See section 2.8.2.48 PortType for further details. Note: current implementation does not support the itemNature International so any use of this element will be ignored.
destinationIs	simple	Optional*	What type of location is receiving the item. See section 2.8.2.49 LocationType for further details.
destinationBusPostalBox	simple	Optional*	Is the receiving location a postal box? See section 2.8.2.1 YesNoOption for further details.
destinationBusUnattended	simple	Optional*	Will the receiving location be unattended? See section 2.8.2.1 YesNoOption for further details.
destinationBusDock	simple	Optional*	Does the receiving location have a dock? See section 2.8.2.1 YesNoOption for further details.
destinationBusForklift	simple	Optional*	Does the receiving location have forklift? See section 2.8.2.1 YesNoOption for further details.
destinationBusLoadingFacilities	simple	Optional*	Does the receiving location have loading facilities? See section 2.8.2.1 YesNoOption for further details.
destinationBusInside	simple	Optional*	Does the receiving location require an inside pickup? See section 2.8.2.1 YesNoOption for further details.
destinationBusNotifyBefore	simple	Optional*	Does the receiving location require notification before pickup? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
destinationBusLimitedAccess	simple	Optional*	Does the receiving location have limited access? See section 2.8.2.1 YesNoOption for further details.
destinationBusHeavyLift	simple	Optional*	Will the item be a heavy lift? See section 2.8.2.1 YesNoOption for further details.

destinationBusTailgateLifter	simple	Optional*	Does the receiving location require a tailgate lifter? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
destinationBusContainerSwingLifter	simple	Optional*	Does the receiving location require a container swing lifter? See section 2.8.2.1 YesNoOption for further details.
destinationResPostalBox	simple	Optional*	Is the receiving location a postal box? See section 2.8.2.1 YesNoOption for further details.
destinationResUnattended	simple	Optional*	Will the receiving location be unattended? See section 2.8.2.1 YesNoOption for further details.
destinationResInside	simple	Optional*	Does the receiving location require an inside pickup? See section 2.8.2.1 YesNoOption for further details.
destinationResNotifyBefore	simple	Optional*	Does the receiving location require notification before pickup? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
destinationResLimitedAccess	simple	Optional*	Does the receiving location have limited access? See section 2.8.2.1 YesNoOption for further details.
destinationResHeavyLift	simple	Optional*	Will the item be a heavy lift? See section 2.8.2.1 YesNoOption for further details.
destinationResTailgateLifter	simple	Optional*	Does the receiving location required a tailgate lifter? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
originIs	simple	Optional*	What type of location is sending the item. See section 2.8.2.49 LocationType for further details.
originBusUnattended	simple	Optional*	Will the sending location be unattended? See section 2.8.2.1 YesNoOption for further details.
originBusDock	simple	Optional*	Does the sending location have a dock? See section 2.8.2.1 YesNoOption for further details.
originBusForklift	simple	Optional*	Does the sending location have forklift? See section 2.8.2.1 YesNoOption for further details.

originBusLoadingFacilities	simple	Optional*	Does the sending location have loading facilities? See section 2.8.2.1 YesNoOption for further details.
originBusInside	simple	Optional*	Does the sending location require an inside pickup? See section 2.8.2.1 YesNoOption for further details.
originBusNotifyBefore	simple	Optional*	Does the sending location require notification before pickup? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
originBusLimitedAccess	simple	Optional*	Does the sending location have limited access? See section 2.8.2.1 YesNoOption for further details.
originBusHeavyLift	simple	Optional*	Will the item be a heavy lift? See section 2.8.2.1 YesNoOption for further details.
originBusTailgateLifter	simple	Optional*	Does the sending location require a tailgate lifter? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
originBusContainerSwingLifter	simple	Optional*	Does the sending location require a container swing lifter? See section 2.8.2.1 YesNoOption for further details.
originResUnattended	simple	Optional*	Will the sending location be unattended? See section 2.8.2.1 YesNoOption for further details.
originResInside	simple	Optional*	Does the sending location require an inside pickup? See section 2.8.2.1 YesNoOption for further details.
originResNotifyBefore	simple	Optional*	Does the sending location require notification before pickup? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does not support this option. Any use of this element will be ignored.
originResLimitedAccess	simple	Optional*	Does the sending location have limited access? See section 2.8.2.1 YesNoOption for further details.
originResHeavyLift	simple	Optional*	Will the item be a heavy lift? See section 2.8.2.1 YesNoOption for further details.
originResTailgateLifter	simple	Optional*	Does the sending location required a tailgate lifter? See section 2.8.2.1 YesNoOption for further details. Note: current implementation does

			<i>not support this option. Any use of this element will be ignored.</i>
--	--	--	--

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
itemMethod	Values will be limited in the following cases: <ul style="list-style-type: none"> If itemNature is "Domestic" options are limited to: Door to Door or Depot to Depot. If itemNature is "International" options are limited to: Door to Door or Port to Port.
originCode	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Door to Door; or itemNature is International and itemMethod is Door to Door.
originSuburb	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Door to Door; or itemNature is International and itemMethod is Door to Door.
originState	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Depot to Depot; or itemNature is International and itemMethod is Port to Port.
originCity	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Depot to Depot; or itemNature is International and itemMethod is Port to Port.
originPort	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is International; and itemMethod is Port to Port.
destinationCode	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Door to Door; or itemNature is International and itemMethod is Door to Door.
destinationSuburb	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Door to Door; or itemNature is International and itemMethod is Door to Door.
destinationState	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Depot to Depot; or itemNature is International and itemMethod is Port to Port.
destinationCity	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is Domestic and itemMethod is Depot to

	Depot; or <ul style="list-style-type: none"> itemNature is International and itemMethod is Port to Port.
destinationPort	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is International; and itemMethod is Port to Port.
portType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemNature is International and itemMethod is Port to Port.
destinationIs	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door.
destinationBusPostalBox	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
destinationBusUnattended	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
destinationBusDock	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusForklift	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusLoadingFacilities	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Boat (see section 2.8.1.2 Anything).
destinationBusInside	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusNotifyBefore	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusLimitedAccess	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusHeavyLift	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is General Goods, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusTailgateLifter	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and

	<ul style="list-style-type: none"> class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationBusContainerSwingLifter	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Business; and class is Freight (see section 2.8.1.2 Anything).
destinationResPostalBox	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
destinationResUnattended	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
destinationResInside	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationResNotifyBefore	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationResLimitedAccess	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
destinationResHeavyLift	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is General Goods, Boat, Animal or Other (see section 2.8.1.2 Anything).
destinationResTailgateLifter	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and destinationIs is Residence; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originIs	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door.
originBusUnattended	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
originBusDock	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusForklift	<p>Applies (and is mandatory) only if:</p> <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).

originBusLoadingFacilities	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Boat (see section 2.8.1.2 Anything).
originBusInside	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusNotifyBefore	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusLimitedAccess	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusHeavyLift	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is General Goods, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusTailgateLifter	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originBusContainerSwingLifter	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Business; and class is Freight (see section 2.8.1.2 Anything).
originResUnattended	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Residence; and class is Freight, General Goods or Other (see section 2.8.1.2 Anything).
originResInside	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Residence; and class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originResNotifyBefore	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Residence; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originResLimitedAccess	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Residence; and class is Freight, General Goods, Vehicle, Boat, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
originResHeavyLift	Applies (and is mandatory) only if: <ul style="list-style-type: none"> itemMethod is Door to Door; and originIs is Residence; and class is General Goods, Boat, Animal or Other (see section 2.8.1.2 Anything).

originResTailgateLifter	Applies (and is mandatory) only if: <ul style="list-style-type: none"> • itemMethod is Door to Door; and • originIs is Residence; and • class is Freight, General Goods, Animal, Refrigerated or Other (see section 2.8.1.2 Anything).
-------------------------	---

Used in: 2.6.1 getQuotesByRequest and 2.6.2 makeBookingByRequest.

Example:

```
<anywhere xsi:type="com:Anywhere">
  <itemNature>Domestic</itemNature>
  <itemMethod>Door to Door</itemMethod>
  <originCountry>AU</originCountry>
  <originCode>4000</originCode>
  <originSuburb>Brisbane</originSuburb>
  <destinationCountry>AU</destinationCountry>
  <destinationCode>2000</destinationCode>
  <destinationSuburb>Sydney</destinationSuburb>
  <destinationIs>Residence</destinationIs>
  <destinationResNotifyBefore>N</destinationResNotifyBefore>
  <destinationResLimitedAccess>N</destinationResLimitedAccess>
  <originIs>Business</originIs>
  <originBusNotifyBefore>Y</originBusNotifyBefore>
  <originBusLimitedAccess>N</originBusLimitedAccess>
</anywhere>
```

2.8.1.4. Anytime

```
<xsd:complexType name="Anytime">
  <xsd:all>
    <xsd:element name="readyDate" type="tns:Date"/>
    <xsd:element name="readyTime" type="tns:ReadyTime"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about when an item/s being sent.

Field	Base type	Occurs	Description
readyDate	simple	Always	What date is the item ready to be sent? See section 2.8.2.50 Date for further details.
readyTime	simple	Always	What time is the item ready to be sent? See section 2.8.2.52 ReadyTime for further details.

Used in: 2.6.1 getQuotesByRequest and 2.6.2 makeBookingByRequest.

Example:

```
<anytime xsi:type="com:Anytime">
  <readyDate>2009-07-01</readyDate>
  <readyTime>PM</readyTime>
</anytime>
```


2.8.1.5. General

```
<xsd:complexType name="General">
  <xsd:all>
    <xsd:element name="goodsValue" type="tns:CurrencyAmount"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

General details about an item/s being sent.

Field	Base type	Occurs	Description
goodsValue	Simple	Optional	The value of the goods being sent. Must be AUD value only. See section 2.8.2.53 CurrencyAmount for further details.

Used in: 2.6.1 getQuotesByRequest and 2.6.2 makeBookingByRequest.

Example:

```
<general xsi:type="com:General">
  <goodsValue>2000.00</goodsValue>
</general>
```

2.8.1.6. AvailableQuote

```
<xsd:complexType name="AvailableQuote">
  <xsd:all>
    <xsd:element name="generated" type="tns:GeneratedType"/>
    <xsd:element name="accepted" type="tns:YesNoOption"/>
    <xsd:element name="bookingNumber" type="tns:BookingNumber"
      minOccurs="0"/>
    <xsd:element name="consignmentNumber" type="tns:ConsignmentNumber"
      minOccurs="0"/>
    <xsd:element name="consignmentDocument"
      type="com:ConsignmentDocument" minOccurs="0"/>
    <xsd:element name="consignmentDocumentType"
      type="com:ConsignmentDocumentType" minOccurs="0"/>
    <xsd:element name="labelDocument"
      type="com:LabelDocument" minOccurs="0"/>
    <xsd:element name="labelDocumentType"
      type="com:LabelDocumentType" minOccurs="0"/>
    <xsd:element name="trackingStatus"
      type="com:TrackingStatus" minOccurs="0"/>
    <xsd:element name="trackingStatusOccurred"
      type="com:Datetime" minOccurs="0"/>
    <xsd:element name="trackingLastChecked"
      type="com:Datetime" minOccurs="0"/>
    <xsd:element name="trackingFurtherDetails"
      type="com:TrackingFurtherDetails" minOccurs="0"/>
    <xsd:element name="totalPrice" type="tns:CurrencyAmount"/>
    <xsd:element name="basePrice" type="tns:CurrencyAmount"/>
    <xsd:element name="tax" type="tns:CurrencyAmount"/>
    <xsd:element name="currency" type="tns:CurrencyType"/>
    <xsd:element name="deliveryMethod" type="tns:DeliveryMethod"/>
    <xsd:element name="usingGeneralRail" type="tns:YesNoOption"/>
    <xsd:element name="usingGeneralRoad" type="tns:YesNoOption"/>
    <xsd:element name="usingGeneralSea" type="tns:YesNoOption"/>
    <xsd:element name="usingExpressAir" type="tns:YesNoOption"/>
  </xsd:all>
</xsd:complexType>
```



```

<xsd:element name="usingExpressRoad" type="tns:YesNoOption"/>
<xsd:element name="etaFrom" type="tns:Eta"/>
<xsd:element name="etaTo" type="tns:Eta"/>
<xsd:element name="guaranteedEta" type="tns:YesNoOption"/>
<xsd:element name="adjustments">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="adjustment" type="tns:Adjustment" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="inclusions">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="inclusion" type="tns:Inclusion" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="extras">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="extra" type="tns:Extra" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="carrier" type="tns:Carrier"/>
<xsd:element name="originatingDepot" type="tns:Depot" minOccurs="0"/>
<xsd:element name="destinationDepot" type="tns:Depot" minOccurs="0"/>
</xsd:all>
<xsd:attribute name="id" type="xsd:positiveInteger" minOccurs="0"/>
</xsd:complexType>

```

Child content:

Specific details about an available quote (a quote that has been accepted).

Field	Base type	Occurs	Description
id (<i>attribute</i>)	positiveInteger	Optional	Attribute: Unique identifier for the quote.
generated	simple	Always	How the quote was generated. See section 2.8.2.86 GeneratedType for further details.
accepted	simple	Always	Whether the quote has been accepted by the client. See section 2.8.2.1 YesNoOption for further details.
bookingNumber	simple	Optional	The carrier booking number representing that the quote has been confirmed by the carrier. See section 2.8.2.75 BookingNumber for further details.
consignmentNumber	simple	Optional	The carrier's consignment number identifies a logical grouping of goods that has been consigned. See section 2.8.2.76 ConsignmentNumber for further details.
consignmentDocument	simple	Optional	The carrier's consignment

			document. Important: The contents of this field are encoded using base64. This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer. When creating a file from this field's contents you will need to ensure that you decode using base64. See section 2.8.2.77 ConsignmentDocument for further details.
consignmentDocumentType	simple	Optional	The type of the carrier's consignment document. This is only provided if an entry for the consignmentDocument field is provided. See section 2.8.2.78 ConsignmentDocumentType for further details.
labelDocument	simple	Optional	The carrier's label document. Important: The contents of this field are encoded using base64. This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer. When creating a file from this field's contents you will need to ensure that you decode using base64. See section 2.8.2.81 LabelDocument for further details.
labelDocumentType	simple	Optional	The type of the carrier's label document. This is only provided if an entry for the labelDocument field is provided. See section 2.8.2.82 LabelDocumentType for further details.
trackingStatus	simple	Optional	The status of the transit of the item/s being sent. See section 2.8.2.83 TrackingStatus for further details.
trackingStatusOccurred	simple	Optional	When the tracking status event occurred. See section 2.8.2.51 Datetime for further details.
trackingLastChecked	simple	Optional	When the tracking status was last checked on. See section 2.8.2.51 Datetime for further details.
trackingFurtherDetails	simple	Optional	Further details on the transit of the item/s being sent. See section 2.8.2.84

			TrackingFurtherDetails for further details.
totalPrice	simple	Always	The total price of the quote, including tax. See section 2.8.2.53 CurrencyAmount for further details.
basePrice	simple	Always	The base price of the quote, before tax is added. See section 2.8.2.53 CurrencyAmount for further details.
tax	simple	Always	The total tax amount for the quote. See section 2.8.2.53 CurrencyAmount for further details.
currency	simple	Always	The currency used for all monetary amounts quoted. See section 2.8.2.54 CurrencyType for further details.
deliveryMethod	simple	Always	The delivery method that the quote price is based on. See section 2.8.2.55 DeliveryMethod for further details.
usingGeneralRail	simple	Always	Whether the delivery method utilises General – Rail as a method of transportation. See section 2.8.2.1 YesNoOption for further details.
usingGeneralRoad	simple	Always	Whether the delivery method utilises General – Road as a method of transportation. See section 2.8.2.1 YesNoOption for further details.
usingGeneralSea	simple	Always	Whether the delivery method utilises General – Sea as a method of transportation. See section 2.8.2.1 YesNoOption for further details.
usingExpressAir	simple	Always	Whether the delivery method utilises Express - Air as a method of transportation. See section 2.8.2.1 YesNoOption for further details.
usingExpressRoad	simple	Always	Whether the delivery method utilises Express - Road as a method of transportation. See section 2.8.2.1 YesNoOption for further details.
etaFrom	simple	Always	The minimum ETA (estimated time of arrival). See section 2.8.2.56 Eta for further details.
etaTo	simple	Always	The maximum ETA (estimated time of arrival).
guaranteedEta	simple	Always	Whether the ETA values supplied are guaranteed by the Carrier. See section 2.8.2.1 YesNoOption for further details.

adjustments	list	Always	Any adjustments to the prices which are applied at the booking stage (eg. credit card fee). See section 2.8.1.7 Adjustment for further details.
inclusions	list	Always	Any inclusions that are included in the price (eg. tracking). See section 2.8.1.10 Inclusion for further details.
extras	list	Always	Any extras that are available for a quote (eg. insurance). See section 2.8.1.11 Extra for further details.
carrier	complex	Always	The carrier that has provided the quote. See section 2.8.1.8 Carrier for further details.
originatingDepot	complex	Optional	Details of the depot that the item is being sent from. Only applies if itemMethod is Depot to Depot (see section 2.8.1.3 Anywhere). See section 2.8.1.9 Depot for further details.
destinationDepot	complex	Optional	Details of the depot that the item is being sent from. Only applies if itemMethod is Depot to Depot (see section 2.8.1.2. Anywhere).

Used in: 2.6.1 getQuotesByRequest.

Example:

```
<quote id="200001" type="com:AvailableQuote">
  <generated>Automatically</generated>
  <accepted>Y</accepted>
  <bookingNumber>ZYX300001</bookingNumber>
  <consignmentNumber>UVW400001</consignmentNumber>
  <consignmentDocument>JVBER ... shortened for readability ... XRhbg
    </consignmentDocument>
  <consignmentDocumentType>application/pdf</consignmentDocumentType>
  <labelDocument>e3Rf3 ... shortened for readability ... X8ggg
    </labelDocument>
  <labelDocumentType>application/pdf</labelDocumentType>
  <trackingStatus>In Transit</trackingStatus>
  <trackingStatusOccurred>2009-07-02T15:25:46Z
    </trackingStatusOccurred>
  <trackingLastChecked>2009-07-02T18:13:12Z</trackingLastChecked>
  <trackingFurtherDetails></trackingFurtherDetails>
  <totalPrice>220.00</totalPrice>
  <basePrice>200.00</basePrice>
  <tax>20.00</tax>
  <currency>AUD</currency>
  <deliveryMethod>Off-peak</deliveryMethod>
  <usingGeneralRail>N</usingGeneralRail>
  <usingGeneralRoad>Y</usingGeneralRoad>
  <usingGeneralSea>N</usingGeneralSea>
  <usingExpressAir>N</usingExpressAir>
  <usingExpressRoad>N</usingExpressRoad>
  <etaFrom>1</etaFrom>
  <etaTo>2</etaTo>
  <guaranteedEta>Y</guaranteedEta>
  <adjustments>
```

```

<adjustment>
  <description>Credit Card Payment Adjustment</description>
  <amount>4.00</amount>
  <tax>0.40</tax>
</adjustment>
<adjustment>
  <description>Credit Payment Adjustment</description>
  <amount>-4.00</amount>
  <tax>-0.40</tax>
</adjustment>
</adjustments>
<inclusions>
  <inclusion>
    <summary>Tracking available</summary>
    <details>Tracking information will be available for this
      quote.</details>
  </inclusion>
  <inclusion>
    <summary>Automatic Booking</summary>
    <details>Automatic booking generated.</details>
  </inclusion>
  <inclusion>
    <summary>Automatic Consignment</summary>
    <details>Automatic consignment note generated.</details>
  </inclusion>
</inclusions>
<extras>
  <extra>
    <summary>Insurance</summary>
    <details>Insurance cover up to $5,000.</details>
    <totalPrice>22.00</totalPrice>
    <basePrice>20.00</basePrice>
    <tax>2.00</tax>
    <mandatory>Y</mandatory>
    <adjustments>
      <adjustment>
        <description>Credit Card Payment Adjustment</description>
        <amount>0.40</amount>
        <tax>0.04</tax>
      </adjustment>
    </adjustments>
  </extra>
</extras>
<carrier>
  <id>654321</id>
  <companyName>ZYX Transport</companyName>
  <companyContact>John Doe</companyContact>
  <streetAddress>1 Transport Street</streetAddress>
  <streetSuburb>Brisbane</streetSuburb>
  <streetCity>Brisbane</streetCity>
  <streetState>QLD</streetState>
  <streetCode>4000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 1111</postalAddress>
  <postalSuburb>Brisbane</postalSuburb>
  <postalCity>Brisbane</postalCity>
  <postalState>QLD</postalState>
  <postalCode>4001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(07) 3333 3333</phone1>
  <email>john.doe@zyxtransport.com.au</email>
  <website>https://www.zyxtransport.com.au</website>
  <conditions>Our conditions include: lots of text.</conditions>
</carrier>
</quote>

```

2.8.1.7. Adjustment

```
<xsd:complexType name="Adjustment">
  <xsd:all>
    <xsd:element name="description" type="tns:AdjustmentDescription"/>
    <xsd:element name="amount" type="tns:CurrencyAmount"/>
    <xsd:element name="tax" type="tns:CurrencyAmount"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about an adjustment. An adjustment is a set of values (ie. amount and tax) that are either added (positive values) or deducted (negative values) from a quote, if the condition for applying the adjustment occurs.

Field	Base type	Occurs	Description
description	simple	Always	A description of the adjustment. See section 2.8.2.57 AdjustmentDescription for further details.
amount	simple	Always	The amount of the adjustment, excluding any tax. See section 2.8.2.53 CurrencyAmount for further details.
tax	simple	Always	The tax amount of the adjustment. See section 2.8.2.53 CurrencyAmount for further details.

Used in: 2.8.1.6 AvailableQuote.

Example:

```
<adjustment xsi:type="com:Adjustment">
  <description>Credit Payment Adjustment</description>
  <amount>-4.00</amount>
  <tax>-0.40</tax>
</adjustment>
```

2.8.1.8. Carrier

```
<xsd:complexType name="Carrier">
  <xsd:all>
    <xsd:element name="id" type="tns:CarrierId"/>
    <xsd:element name="companyName" type="tns:CompanyName"/>
    <xsd:element name="companyContact" type="tns:ContactName"/>
    <xsd:element name="streetAddress" type="tns:Address" minOccurs="0"/>
    <xsd:element name="streetSuburb" type="tns:Suburb" minOccurs="0"/>
    <xsd:element name="streetCity" type="tns:City" minOccurs="0"/>
    <xsd:element name="streetState" type="tns:State" minOccurs="0"/>
    <xsd:element name="streetCode" type="tns:PostalCode" minOccurs="0"/>
    <xsd:element name="streetCountry" type="tns:CountryCode" minOccurs="0"/>
    <xsd:element name="postalAddress" type="tns:Address" minOccurs="0"/>
    <xsd:element name="postalSuburb" type="tns:Suburb" minOccurs="0"/>
    <xsd:element name="postalCity" type="tns:City" minOccurs="0"/>
    <xsd:element name="postalState" type="tns:State" minOccurs="0"/>
    <xsd:element name="postalCode" type="tns:PostalCode" minOccurs="0"/>
    <xsd:element name="postalCountry" type="tns:CountryCode" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

```

<xsd:element name="phone1" type="tns:Phone" minOccurs="0"/>
<xsd:element name="phone2" type="tns:Phone" minOccurs="0"/>
<xsd:element name="email" type="tns:Email" minOccurs="0"/>
<xsd:element name="website" type="tns:Website" minOccurs="0"/>
<xsd:element name="conditions" type="tns:CarrierConditions"
  minOccurs="0"/>
</xsd:all>
</xsd:complexType>

```

Child content:

Specific details about a carrier.

Field	Base type	Occurs	Description
carrierId	simple	Always	The unique identifier for a carrier. See section 2.8.2.58 CarrierId for further details.
companyName	simple	Always	The name of a carrier. See section 2.8.2.59 CompanyName for further details.
companyContact	simple	Always	The person's name or position description of the contact for the carrier. See section 2.8.2.61 ContactName for further details.
streetAddress	simple	Optional	The street details for the street address of the carrier. See section 2.8.2.37 Address for further details.
streetSuburb	simple	Optional	The suburb for the street address of the carrier. See section 2.8.2.38 Suburb for further details.
streetCity	simple	Optional	The city for the street address of the carrier. See section 2.8.2.40 City for further details.
streetState	simple	Optional	The state for the street address of the carrier. See section 2.8.2.39 State for further details.
streetCode	simple	Optional	The postal code for the street address of the carrier. See section 2.8.2.41 PostalCode for further details.
streetCountry	simple	Optional	The country for the street address of the carrier. See section 2.8.2.42 CountryCode for further details.
postalAddress	simple	Optional	The street details for the postal address of the carrier. See section 2.8.2.37 Address for further details.
postalSuburb	simple	Optional	The suburb for the postal address of the carrier. See section 2.8.2.38 Suburb for further details.
postalCity	simple	Optional	The city for the postal address of the carrier. See section 2.8.2.40 City for

			further details.
postalState	simple	Optional	The state for the postal address of the carrier. See section 2.8.2.39 State for further details.
postalCode	simple	Optional	The postal code for the postal address of the carrier. See section 2.8.2.41 PostalCode for further details.
postalCountry	simple	Optional	The country for the postal address of the carrier. See section 2.8.2.42 CountryCode for further details.
phone1	simple	Optional	The first phone number for the carrier. See section 2.8.2.43 Phone for further details.
phone2	simple	Optional	The second phone number for the carrier. See section 2.8.2.43 Phone for further details.
Email	simple	Optional	The contact email address for the carrier. See section 2.8.2.45 Email for further details.
Website	simple	Optional	The carrier's website. See section 2.8.2.46 Website for further details.
Conditions	simple	Optional	The carrier specific conditions for transporting an item/s. See section 2.8.2.62 CarrierConditions for further details.

Used in: 2.8.1.6 AvailableQuote.

Example:

```
<carrier xsi:type="com:Carrier">
  <id>654321</id>
  <companyName>ZYX Transport</companyName>
  <companyContact>John Doe</companyContact>
  <streetAddress>1 Transport Street</streetAddress>
  <streetSuburb>Brisbane</streetSuburb>
  <streetCity>Brisbane</streetCity>
  <streetState>QLD</streetState>
  <streetCode>4000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 1111</postalAddress>
  <postalSuburb>Brisbane</postalSuburb>
  <postalCity>Brisbane</postalCity>
  <postalState>QLD</postalState>
  <postalCode>4001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(07) 3333 3333</phone1>
  <email>john.doe@zyxtransport.com.au</email>
  <website>https://www.zyxtransport.com.au</website>
  <conditions>Our conditions include: lots of text.</conditions>
</carrier>
```


2.8.1.9. Depot

```
<xsd:complexType name="Depot">
  <xsd:all>
    <xsd:element name="name" type="tns:DepotName"/>
    <xsd:element name="street" type="tns:Address" minOccurs="0"/>
    <xsd:element name="suburb" type="tns:Suburb" minOccurs="0"/>
    <xsd:element name="city" type="tns:City" minOccurs="0"/>
    <xsd:element name="state" type="tns:State" minOccurs="0"/>
    <xsd:element name="code" type="tns:PostalCode" minOccurs="0"/>
    <xsd:element name="country" type="tns:CountryCode" minOccurs="0"/>
    <xsd:element name="phone1" type="tns:Phone" minOccurs="0"/>
    <xsd:element name="phone2" type="tns:Phone" minOccurs="0"/>
    <xsd:element name="fax" type="tns:Fax" minOccurs="0"/>
    <xsd:element name="instructions" type="tns:DepotInstructions"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about a carrier's depot.

Field	Base type	Occurs	Description
name	simple	Always	The name of the depot, as defined by the carrier. See section 2.8.2.63 DepotName for further details.
street	simple	Optional	The street details for the street address of the carrier. See section 2.8.2.37 Address for further details.
suburb	simple	Optional	The suburb for the street address of the carrier. See section 2.8.2.38 Suburb for further details.
city	simple	Optional	The city for the street address of the carrier. See section 2.8.2.40 City for further details.
state	simple	Optional	The state for the street address of the carrier. See section 2.8.2.39 State for further details.
code	simple	Optional	The postal code for the street address of the carrier. See section 2.8.2.41 PostalCode for further details.
country	simple	Optional	The country for the street address of the carrier. See section 2.8.2.42 CountryCode for further details.
phone1	simple	Optional	The first phone number for the carrier. See section 2.8.2.43 Phone for further details.
phone2	simple	Optional	The second phone number for the carrier. See section 2.8.2.43 Phone for further details.
fax	simple	Optional	The fax number for the carrier.

			See section 2.8.2.44 Fax for further details.
instructions	simple	Optional	Any specific instructions relating to the drop-off or pickup of items to/from this depot. See section 2.8.2.64 DepotInstructions for further details.

Used in: 2.8.1.6 AvailableQuote.

Example:

```
<depot xsi:type="com:Depot">
  <name>BNE Depot</name>
  <street>1 Transport Street</streetAddress>
  <suburb>Brisbane</streetSuburb>
  <city>Brisbane</streetCity>
  <state>QLD</streetState>
  <code>4000</streetCode>
  <country>AU</streetCountry>
  <phone1>(07) 3333 3333</phone1>
  <fax>(07) 3333 3331</fax>
  <instructions>Come to the Red door.</instructions>
</depot>
```

2.8.1.10. Inclusion

```
<xsd:complexType name="Inclusion">
  <xsd:all>
    <xsd:element name="summary" type="tns:InclusionSummary"/>
    <xsd:element name="details" type="tns:InclusionDetails"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about an inclusion for a quote.

Field	Base type	Occurs	Description
summary	simple	Always	The summary of the inclusion details. This is a unique identifier for the inclusion. See section 2.8.2.65 InclusionSummary for further details.
details	simple	Always	The full details of the inclusion. See section 2.8.2.66 InclusionDetails for further details.

Used in: 2.8.1.6 AvailableQuote.

Example:

```
<inclusion xsi:type="com:Inclusion">
  <summary>Tracking available</summary>
  <details>Tracking information will be available for this quote.
  </details>
</inclusion>
```

2.8.1.11. Extra

```
<xsd:complexType name="Extra">
  <xsd:all>
    <xsd:element name="summary" type="tns:ExtraSummary"/>
    <xsd:element name="details" type="tns:ExtraDetails" minOccurs="0"/>
    <xsd:element name="totalPrice" type="tns:CurrencyAmount"/>
    <xsd:element name="basePrice" type="tns:CurrencyAmount"/>
    <xsd:element name="tax" type="tns:CurrencyAmount"/>
    <xsd:element name="mandatory" type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="adjustments">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="adjustment" type="tns:Adjustment" minOccurs="0"
            maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about an extra available for a quote.

Field	Base type	Occurs	Description
summary	simple	Always	The summary of the inclusion details. This is a unique identifier for the extra. See section 2.8.2.67 ExtraSummary for further details.
details	simple	Optional	The full details of the inclusion. See section 2.8.2.68 ExtraDetails for further details.
totalPrice	simple	Always	The total price of the extra, including tax. See section 2.8.2.53 CurrencyAmount for further details.
basePrice	simple	Always	The base price of the extra, before tax is added. See section 2.8.2.53 CurrencyAmount for further details.
tax	simple	Always	The total tax amount for the extra. See section 2.8.2.53 CurrencyAmount for further details.
mandatory	simple	Optional	Whether the extra is mandatory or not. See section 2.8.2.1 YesNoOption for further details.
adjustments	list	Always	Any adjustments to the prices which are applied at the booking stage (eg. credit card fee). See section 2.8.1.7 Adjustment for further details.

Used in: 2.8.1.6 AvailableQuote and 2.8.1.14 BookingQuote.

Example:

```
<extra xsi:type="com:Extra">
  <summary>Insurance</summary>
  <details>Insurance cover up to $5,000.</details>
  <totalPrice>22.00</totalPrice>
  <basePrice>20.00</basePrice>
  <tax>2.00</tax>
  <mandatory>Y</mandatory>
  <adjustments>
    <adjustment>
      <description>Credit Card Payment Adjustment</description>
      <amount>0.40</amount>
      <tax>0.04</tax>
    </adjustment>
  </adjustments>
</extra>
```

2.8.1.12. Location

```
<xsd:complexType name="Location">
  <xsd:all>
    <xsd:element name="description" type="tns:LocationName" minOccurs="0"/>
    <xsd:element name="type" type="tns:LocationPosition" minOccurs="0"/>
    <xsd:element name="contactName" type="tns:ContactName" minOccurs="0"/>
    <xsd:element name="companyName" type="tns:CompanyName" minOccurs="0"/>
    <xsd:element name="street" type="tns:Address" minOccurs="0"/>
    <xsd:element name="suburb" type="tns:Suburb" minOccurs="0"/>
    <xsd:element name="state" type="tns:State" minOccurs="0"/>
    <xsd:element name="code" type="tns:PostalCode" minOccurs="0"/>
    <xsd:element name="country" type="tns:CountryCode" minOccurs="0"/>
    <xsd:element name="phone1" type="tns:Phone" minOccurs="0"/>
    <xsd:element name="phone2" type="tns:Phone" minOccurs="0"/>
    <xsd:element name="fax" type="tns:Fax" minOccurs="0"/>
    <xsd:element name="email" type="tns:Email" minOccurs="0"/>
    <xsd:element name="postalBox" type="tns:YesNoOption" minOccurs="0"/>
    <xsd:element name="dock" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="forklift" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="loadingFacilities" type="tns:YesNoOption"
      minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

The details about where an item/s is being sent from or the originating address.

Field	Base type	Occurs	Description
description	simple	Optional*	A description allocated to a predefined location stored in the client's location listings. See section 2.8.2.60 LocationName for further details.
type	simple	Optional*	Whether the location is a originating address (ie. pickup) or a destination address (dropoff) See section 2.8.2.100 LocationPosition for further

			details.
contactName	simple	Optional*	A contact name for the location. See section 2.8.2.61 ContactName for further details.
companyName	simple	Optional*	A descriptive name for the location. See section 2.8.2.59 CompanyName for further details.
street	simple	Optional*	The street details for the address. See section 2.8.2.37 Address for further details.
suburb	simple	Optional*	The suburb for the address. See section 2.8.2.38 Suburb for further details.
state	simple	Optional*	The state for the address. See section 2.8.2.39 State for further details.
code	simple	Optional*	The postal code for the address. See section 2.8.2.41 PostalCode for further details.
country	simple	Optional*	The country for the address. See section 2.8.2.42 CountryCode for further details.
phone1	simple	Optional*	The first phone number for the address. See section 2.8.2.43 Phone for further details.
phone2	simple	Optional	The second phone number for the address. See section 2.8.2.43 Phone for further details.
fax	simple	Optional	The fax number for the address. See section 2.8.2.44 Fax for further details.
email	simple	Optional*	The email address for the address. See section 2.8.2.45 Email for further details.
postalBox	simple	Optional*	Is the address a postal box? See section 2.8.2.1 YesNoOption for further details.
dock	simple	Optional*	Does the address have a dock? See section 2.8.2.1 YesNoOption for further details.
forklift	simple	Optional*	Does the address have a forklift? See section 2.8.2.1 YesNoOption for further details.
loadingFacilities	simple	Optional*	Does the address have loading facilities? See section 2.8.2.1 YesNoOption for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
contactName	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
type	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
companyName	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied; and originIs/destinationIs is Business (see section 2.8.1.3 Anywhere).
street	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
suburb	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
state	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
code	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
country	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
phone1	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
email	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
postalBox	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
dock	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
forklift	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.
loadingFacilities	Applies (and is mandatory) only if: <ul style="list-style-type: none"> description has not been supplied.

Used in: 2.5.1 Get Quotes and Make a Booking (Client User), 2.6.2 makeBookingByRequest and 2.8.1.1 Request.

Example:

```
<location xsi:type="com:Location">
  <contactName>John Doe</contactName>
  <type>Origin</type>
  <companyName>ABC Suppliers</companyName>
  <street>1 Originating Street</street>
  <suburb>Brisbane</suburb>
  <state>QLD</state>
  <code>4000</code>
  <country>AU</country>
  <phone1>(07) 3333 3333</phone1>
  <phone2></phone2>
  <fax>(07) 3333 3331</fax>
  <email>bneoffice@abcsuppliers.com.au</email>
  <postalBox>N</postalBox>
  <dock>N</dock>
  <forklift>N</forklift>
  <loadingFacilities>N</loadingFacilities>
</location>
```

2.8.1.13. Payment

```
<xsd:complexType name="Payment">
  <xsd:all>
    <xsd:element name="paymentType" type="tns:PaymentType"/>
    <xsd:element name="cardType" type="tns:CreditCardType" minOccurs="0"/>
    <xsd:element name="cardExpiryDate" type="tns:CreditCardExpiryDate"
      minOccurs="0"/>
    <xsd:element name="cardNumber" type="tns:CreditCardNumber"
      minOccurs="0"/>
    <xsd:element name="cardName" type="tns:CreditCardName" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

The details about a payment. Includes information about the payment type and any specific details related to the payment type specified.

Field	Base type	Occurs	Description
paymentType	simple	Always	The payment method to be used for paying an amount. See section 2.8.2.69 PaymentType for further details.
cardType	simple	Optional	The type of credit card being supplied. See section 2.8.2.70 CreditCardType for further details.
cardExpiryDate	simple	Optional	The expiry date of the credit card being supplied. See section 2.8.2.71 CreditCardExpiryDate for further details.
cardNumber	simple	Optional	The number of the credit card being supplied. See section 2.8.2.72 CreditCardNumber for further details.
cardName	simple	Optional	The name of the owner of the credit card being supplied. See section 2.8.2.73 CreditCardName for further details.

Further restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
cardType	Applies (and is mandatory) only if: <ul style="list-style-type: none"> paymentType is Credit Card.
cardExpiryDate	1. Applies (and is mandatory) only if: <ul style="list-style-type: none"> paymentType is Credit Card.

	2. Must be a date after the current date.
cardNumber	1. Applies (and is mandatory) only if: <ul style="list-style-type: none"> • paymentType is Credit Card. 2. Validated based on rules specific to cardTypes and using the Luhn algorithm (mod 10).
cardName	Applies (and is mandatory) only if: <ul style="list-style-type: none"> • paymentType is Credit Card.

Used in: 2.6.2 makeBookingByRequest.

Example:

```
<payment xsi:type="com:Payment">
  <paymentType>Credit Card</paymentType>
  <cardType>MasterCard</cardType>
  <cardExpiryDate>05-2010</cardExpiryDate>
  <cardNumber>5163190000000000</cardNumber>
  <cardName>JANE DOE</cardName>
</payment>
```

2.8.1.14. BookingQuote

```
<xsd:complexType name="BookingQuote">
  <xsd:all>
    <xsd:element name="totalPrice" type="tns:CurrencyAmount"/>
    <xsd:element name="basePrice" type="tns:CurrencyAmount"/>
    <xsd:element name="tax" type="tns:CurrencyAmount"/>
    <xsd:element name="currency" type="tns:CurrencyType"/>
    <xsd:element name="deliveryMethod" type="tns:DeliveryMethod"/>
    <xsd:element name="etaFrom" type="tns:Eta"/>
    <xsd:element name="etaTo" type="tns:Eta"/>
    <xsd:element name="guaranteedEta" type="tns:YesNoOption"/>
    <xsd:element name="carrierId" type="tns:CarrierId"/>
    <xsd:element name="extras">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="extra" type="tns:Extra" minOccurs="0"
            maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
```

Child content:

Specific details about a quote to be accepted (a quote that has not yet been accepted).

Field	Base type	Occurs	Description
totalPrice	simple	Always	The total price of the quote, including tax. See section 2.8.2.53 CurrencyAmount for further details.
basePrice	simple	Always	The base price of the quote, before tax is added. See section 2.8.2.53 CurrencyAmount for further details.

tax	simple	Always	The total tax amount for the quote. See section 2.8.2.53 CurrencyAmount for further details.
currency	simple	Always	The currency used for all monetary amounts quoted. See section 2.8.2.54 CurrencyType for further details.
deliveryMethod	simple	Always	The delivery method that the quote price is based on. See section 2.8.2.55 DeliveryMethod for further details.
etaFrom	simple	Always	The minimum ETA (estimated time of arrival). See section 2.8.2.56 Eta for further details.
etaTo	simple	Always	The maximum ETA (estimated time of arrival). See section 2.8.2.56 Eta for further details.
guaranteedEta	simple	Always	Whether the ETA values supplied are guaranteed by the carrier. See section 2.8.2.1 YesNoOption for further details.
carrierId	simple	Always	The unique identifier of a carrier providing the quote. See section 2.8.2.58 CarrierId for further details.
extras	list	Optional	Any extras to be included for the quote (eg. insurance). See section 2.8.1.11 Extra for further details.

Used in: 2.6.2 makeBookingByRequest.

Example:

```
<quote xsi:type="com:BookingQuote">
  <totalPrice>110.00</totalPrice>
  <basePrice>100.00</basePrice>
  <tax>10.00</tax>
  <currency>AUD</currency>
  <deliveryMethod>Off-peak</deliveryMethod>
  <etaFrom>2</etaFrom>
  <etaTo>3</etaTo>
  <guaranteedEta>Y</guaranteedEta>
  <carrierId>654321</carrierId>
  <extras>
    <extra>
      <summary>Insurance</summary>
      <totalPrice>22.00</totalPrice>
      <basePrice>20.00</basePrice>
      <tax>2.00</tax>
    </extra>
  </extras>
</quote>
```

2.8.1.15. Client

```
<xsd:complexType name="Client">
```

```

<xsd:all>
  <xsd:element name="id" type="tns:ClientId" minOccurs="0"/>
  <xsd:element name="individualCompany" type="tns:IndividualCompany"/>
  <xsd:element name="companyName" type="tns:CompanyName" minOccurs="0"/>
  <xsd:element name="companyContactPerson" type="tns:ContactName"
    minOccurs="0"/>
  <xsd:element name="companyNo" type="tns:CompanyNumber"
    minOccurs="0"/>
  <xsd:element name="individualSurname" type="tns:Surname"
    minOccurs="0"/>
  <xsd:element name="individualFirstname" type="tns:Firstname"
    minOccurs="0"/>
  <xsd:element name="individualDateOfBirth" type="tns:Date"
    minOccurs="0"/>
  <xsd:element name="streetAddress" type="tns:Address" minOccurs="0"/>
  <xsd:element name="streetSuburb" type="tns:Suburb" minOccurs="0"/>
  <xsd:element name="streetCity" type="tns:City" minOccurs="0"/>
  <xsd:element name="streetState" type="tns:State" minOccurs="0"/>
  <xsd:element name="streetCode" type="tns:PostalCode" minOccurs="0"/>
  <xsd:element name="streetCountry" type="tns:CountryCode"/>
  <xsd:element name="postalAddress" type="tns:Address" minOccurs="0"/>
  <xsd:element name="postalSuburb" type="tns:Suburb" minOccurs="0"/>
  <xsd:element name="postalCity" type="tns:City" minOccurs="0"/>
  <xsd:element name="postalState" type="tns:State" minOccurs="0"/>
  <xsd:element name="postalCode" type="tns:PostalCode" minOccurs="0"/>
  <xsd:element name="postalCountry" type="tns:CountryCode"
    minOccurs="0"/>
  <xsd:element name="phone1" type="tns:Phone"/>
  <xsd:element name="phone2" type="tns:Phone" minOccurs="0"/>
  <xsd:element name="email" type="tns:Email"/>
</xsd:all>
</xsd:complexType>

```

Child content:

Specific details about a client.

Field	Base type	Occurs	Description
id	simple	Optional*	The unique identifier for a client. See section 2.8.2.93 ClientId for further details.
individualCompany	simple	Always	Whether the account is for an individual or a company. See section 2.8.2.89 IndividualCompany for further details.
companyName	simple	Optional*	The company name of a client. See section 2.8.2.59 CompanyName for further details.
companyContactPerson	simple	Optional*	The person's name or position description of the contact for the client. See section 2.8.2.61 ContactName for further details.
companyNo	simple	Optional*	The allocated company number for a client. See section 2.8.2.92 CompanyNumber for further details.
individualSurname	simple	Optional*	The individual client's surname. See section 2.8.2.90 Surname for further details.

individualFirstname	simple	Optional*	The individual client's firstname. See section 2.8.2.91 Firstname for further details.
individualDateOfBirth	simple	Optional*	The individual client's date of birth. See section 2.8.2.50 Date for further details.
streetAddress	simple	Optional	The street details for the street address of the client. See section 2.8.2.37 Address for further details.
streetSuburb	simple	Optional	The suburb for the street address of the client. See section 2.8.2.38 Suburb for further details.
streetCity	simple	Optional	The city for the street address of the client. See section 2.8.2.40 City for further details.
streetState	simple	Optional	The state for the street address of the client. See section 2.8.2.39 State for further details.
streetCode	simple	Optional	The postal code for the street address of the client. See section 2.8.2.41 PostalCode for further details.
streetCountry	simple	Always	The country for the street address of the client. See section 2.8.2.42 CountryCode for further details.
postalAddress	simple	Optional	The street details for the postal address of the client. See section 2.8.2.37 Address for further details.
postalSuburb	simple	Optional	The suburb for the postal address of the client. See section 2.8.2.38 Suburb for further details.
postalCity	simple	Optional	The city for the postal address of the client. See section 2.8.2.40 City for further details.
postalState	simple	Optional	The state for the postal address of the client. See section 2.8.2.39 State for further details.
postalCode	simple	Optional	The postal code for the postal address of the client. See section 2.8.2.41 PostalCode for further details.
postalCountry	simple	Optional	The country for the postal address of the client. See section 2.8.2.42 CountryCode for further details.
phone1	simple	Optional	The first phone number for the client. See section 2.8.2.43 Phone for further details.

phone2	simple	Optional	The second phone number for the client. See section 2.8.2.43 Phone for further details.
Email	simple	Optional	The contact email address for the client. See section 2.8.2.45 Email for further details.

* although the Occurs is specified in the XML definition further restrictions are applied to this field. Detailed in Further Restrictions section below.

Further restrictions:

Further to the restrictions applied by the field's base type and element conditions (eg. minOccurs="0") the following rules will also be applied after the data has passed the XML tests:

Field	Further restrictions
id	Applies (and is mandatory) only if: <ul style="list-style-type: none"> the client already exists.
companyName	Applies (and is mandatory) only if: <ul style="list-style-type: none"> individualCompany is Company.
companyContactPerson	Applies (and is mandatory) only if: <ul style="list-style-type: none"> individualCompany is Company.
companyNo	Applies (and is mandatory) only if: <ul style="list-style-type: none"> individualCompany is Company.
individualSurname	Applies (and is mandatory) only if: <ul style="list-style-type: none"> individualCompany is Individual.
individualFirstname	Applies (and is mandatory) only if: <ul style="list-style-type: none"> individualCompany is Individual.
individualDateOfBirth	Applies only if: <ul style="list-style-type: none"> individualCompany is Individual.

Used in: 2.6.7 createClient, 2.6.8 updateClient and 2.6.9 getClient.

Example:

```
<client xsi:type="com:Client">
  <id>123456</id>
  <individualCompany>Individual</individualCompany>
  <individualSurname>Smith</individualSurname>
  <individualFirstname>John</individualFirstname>
  <individualDateOfBirth>1980-04-21</individualDateOfBirth>
  <streetAddress>1 User Street</streetAddress>
  <streetSuburb>Melbourne</streetSuburb>
  <streetCity>Melbourne</streetCity>
  <streetState>VIC</streetState>
  <streetCode>3000</streetCode>
  <streetCountry>AU</streetCountry>
  <postalAddress>PO Box 3333</postalAddress>
  <postalSuburb>Melbourne</postalSuburb>
  <postalCity>Melbourne</postalCity>
  <postalState>VIC</postalState>
  <postalCode>3001</postalCode>
  <postalCountry>AU</postalCountry>
  <phone1>(03) 9999 9999</phone1>
  <email>john.smith@mailhost.com.au</email>
</client>
```

2.8.1.16. LoginDetails

```
<xsd:complexType name="LoginDetails">
  <xsd:all>
    <xsd:element name="loginId" type="tns:LoginId"/>
    <xsd:element name="password" type="tns:Password"/>
  </xsd:all>
</xsd:complexType>
```

Child content:

The login details for an account.

Field	Base type	Occurs	Description
loginId	simple	Always	Login id for an account. See section 2.8.2.87 LoginId for further details.
password	simple	Always	Password for an account. See section 2.8.2.88 Password for further details.

Used in: 2.6.7 createClient and 2.6.9 getClient.

Example:

```
<loginDetails xsi:type="com:LoginDetails">
  <loginId>jsmith</loginId>
  <password>my_password1357</password>
</loginDetails>
```

2.8.1.17. DispatchDetails

```
<xsd:complexType name="DispatchDetails">
  <xsd:all>
    <xsd:element name="reference" type="tns:CarrierReference"
      minOccurs="0"/>
    <xsd:element name="carrierName" type="tns:CompanyName" minOccurs="0"/>
    <xsd:element name="carrierId" type="tns:CarrierId" minOccurs="0"/>
    <xsd:element name="accountNo" type="tns:CarrierAccountNumber"
      minOccurs="0"/>
    <xsd:element name="creatorId" type="tns:CarrierCreatorId"
      minOccurs="0"/>
    <xsd:element name="currency" type="tns:CurrencyType" minOccurs="0"/>
    <xsd:element name="changedCarrier" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="consignmentEdited" type="tns:YesNoOption"
      minOccurs="0"/>
    <xsd:element name="consignmentNumber" type="tns:ConsignmentNumber"
      minOccurs="0"/>
    <xsd:element name="consignmentDate" type="tns:Date" minOccurs="0"/>
    <xsd:element name="deliveryMethod" type="tns:DeliveryMethod"
      minOccurs="0"/>
    <xsd:element name="rateName" type="tns:RateName" minOccurs="0"/>
    <xsd:element name="description" type="tns:ItemDescription"
      minOccurs="0"/>
    <xsd:element name="distanceMeasurementType"
      type="tns:DistanceMeasurementType" minOccurs="0"/>
    <xsd:element name="weightMeasurementType"
      type="tns:WeightMeasurementType" minOccurs="0"/>
    <xsd:element name="length" type="tns:Length" minOccurs="0"/>
    <xsd:element name="width" type="tns:Width" minOccurs="0"/>
    <xsd:element name="height" type="tns:Height" minOccurs="0"/>
    <xsd:element name="actualWeight" type="tns:Weight" minOccurs="0"/>
  </xsd:all>
</xsd:complexType>
```

```

<xsd:element name="chargeableweight" type="tns:Weight"
  minOccurs="0"/>
<xsd:element name="actualCubic" type="tns:Cubic" minOccurs="0"/>
<xsd:element name="chargeableCubic" type="tns:Cubic"
  minOccurs="0"/>
<xsd:element name="actualQuantity" type="tns:Quantity" minOccurs="0"/>
<xsd:element name="chargeableQuantity" type="tns:Quantity"
  minOccurs="0"/>
<xsd:element name="basePrice" type="tns:CurrencyAmount"
  minOccurs="0"/>
<xsd:element name="surcharges" type="tns:CurrencyAmount"
  minOccurs="0"/>
<xsd:element name="tax" type="tns:CurrencyAmount" minOccurs="0"/>
<xsd:element name="totalPrice" type="tns:CurrencyAmount"
  minOccurs="0"/>
<xsd:element name="originZoneName" type="tns:ZoneName" minOccurs="0"/>
<xsd:element name="originContactName" type="tns:ContactName"
  minOccurs="0"/>
<xsd:element name="originCompanyName" type="tns:CompanyName"
  minOccurs="0"/>
<xsd:element name="originStreet" type="tns:Address" minOccurs="0"/>
<xsd:element name="originSuburb" type="tns:Suburb" minOccurs="0"/>
<xsd:element name="originState" type="tns:State" minOccurs="0"/>
<xsd:element name="originCode" type="tns:PostalCode" minOccurs="0"/>
<xsd:element name="originCountry" type="tns:CountryCode"
  minOccurs="0"/>
<xsd:element name="originPhone1" type="tns:Phone" minOccurs="0"/>
<xsd:element name="originPhone2" type="tns:Phone" minOccurs="0"/>
<xsd:element name="originFax" type="tns:Fax" minOccurs="0"/>
<xsd:element name="originEmail" type="tns:Email" minOccurs="0"/>
<xsd:element name="destinationZoneName" type="tns:ZoneName"
  minOccurs="0"/>
<xsd:element name="destinationContactName" type="tns:ContactName"
  minOccurs="0"/>
<xsd:element name="destinationCompanyName" type="tns:CompanyName"
  minOccurs="0"/>
<xsd:element name="destinationStreet" type="tns:Address"
  minOccurs="0"/>
<xsd:element name="destinationSuburb" type="tns:Suburb"
  minOccurs="0"/>
<xsd:element name="destinationState" type="tns:State" minOccurs="0"/>
<xsd:element name="destinationCode" type="tns:PostalCode"
  minOccurs="0"/>
<xsd:element name="destinationCountry" type="tns:CountryCode"
  minOccurs="0"/>
<xsd:element name="destinationPhone1" type="tns:Phone" minOccurs="0"/>
<xsd:element name="destinationPhone2" type="tns:Phone" minOccurs="0"/>
<xsd:element name="destinationFax" type="tns:Fax" minOccurs="0"/>
<xsd:element name="destinationEmail" type="tns:Email" minOccurs="0"/>
</xsd:all>
</xsd:complexType>

```

Child content:

The dispatch details of the consignment.

Field	Base type	Occurs	Description
reference	simple	Optional	The reference code used by a carrier company to refer to a particular request See section 2.8.2.94 CarrierReference for further details.
carrierName	simple	Optional	The name of the carrier sending the item/s. See section 2.8.2.59 CompanyName for further details.

carrierId	simple	Optional	The unique id for a carrier. See section 2.8.2.58 CarrierId for further details.
accountNo	simple	Optional	The reference code used by a carrier company to refer to the Temando account. See section 2.8.2.95 CarrierAccountNumber for further details.
creatorId	simple	Optional	The reference code used by a carrier company to refer to the third party (ie. not Temando) that created a request. See section 2.8.2.96 CarrierCreatorId for further details.
currency	simple	Optional	The currency used for all monetary amounts. See section 2.8.2.54 CurrencyType for further details.
changedCarrier	simple	Optional	Whether the originally accepted quote has been changed to another carrier. See section 2.8.2.1 YesNoOption for further details.
consignmentEdited	simple	Optional	Whether the originally generated consignment has been changed. See section 2.8.2.1 YesNoOption for further details.
consignmentNumber	simple	Optional	The carrier's consignment number identifies a logical grouping of goods that has been consigned. See section 2.8.2.76 ConsignmentNumber for further details.
consignmentDate	simple	Optional	When the consignment was generated. See section 2.8.2.50 Date for further details.
deliveryMethod	simple	Optional	The delivery method to be used for the request. See section 2.8.2.55 DeliveryMethod for further details.
rateName	simple	Optional	The rate classification for the item/s that was used for rate calculation. See section 2.8.2.97 RateName for further details.
description	simple	Optional	Further details of the item being sent, that have not been specified in other elements. See section 2.8.2.32 ItemDescription for further details.
distanceMeasurementType	simple	Optional	The distance measurement type to be used for the Length, Width and Height values. See section 2.8.2.25 DistanceMeasurementType for further details.
weightMeasurementType	simple	Optional	The weight measurement type to be used for the Weight values. See section 2.8.2.26 WeightMeasurementType for further details.
length	simple	Optional	The length of the item being sent. See section 2.8.2.27 Length for further details.

width	simple	Optional	The width of the item being sent. See section 2.8.2.28 Width for further details.
height	simple	Optional	The height of the item being sent. See section 2.8.2.29 Height for further details.
actualWeight	simple	Optional	The actual weight of the item being sent. See section 2.8.2.30 Weight for further details.
chargeableWeight	simple	Optional	The chargeable weight of the item being sent. See section 2.8.2.30 Weight for further details.
actualCubic	simple	Optional	The actual cubic of the item being sent. See section 2.8.2.98 Cubic for further details.
chargeableCubic	simple	Optional	The chargeable cubic of the item being sent. See section 2.8.2.98 Cubic for further details.
actualQuantity	simple	Optional	The actual number of items being sent. See section 2.8.2.31 Quantity for further details.
chargeableQuantity	simple	Optional	The chargeable number of items being sent. See section 2.8.2.31 Quantity for further details.
basePrice	simple	Optional	The base price of the quote, excluding surcharges and tax. See section 2.8.2.53 CurrencyAmount for further details.
surcharges	simple	Optional	Any surcharges to be included in the total price. See section 2.8.2.53 CurrencyAmount for further details.
tax	simple	Optional	The tax amount. See section 2.8.2.53 CurrencyAmount for further details.
totalPrice	simple	Optional	The total price of the quote, including base price, surcharges and tax. See section 2.8.2.53 CurrencyAmount for further details.
originZoneName	simple	Optional	A name of the carrier zone that the sender's address is matched to. See section 2.8.2.99 ZoneName for further details.
originContactName	simple	Optional	A contact name for an individual at the origin location. See section 2.8.2.61 ContactName for further details.
originCompanyName	simple	Optional	The company name (if relevant) for the sender. See section 2.8.2.59 CompanyName for further details.
originStreet	simple	Optional	The street details for the address. See section 2.8.2.37 Address for further details.
originSuburb	simple	Optional	The suburb for the address. See section 2.8.2.38 Suburb for further details.

originState	simple	Optional	The state for the address. See section 2.8.2.39 State for further details.
originCode	simple	Optional	The postal code for the address. See section 2.8.2.41 PostalCode for further details.
originCountry	simple	Optional	The country for the address. See section 2.8.2.42 CountryCode for further details.
originPhone1	simple	Optional	The first phone number for the address. See section 2.8.2.43 Phone for further details.
originPhone2	simple	Optional	The second phone number for the address. See section 2.8.2.43 Phone for further details.
originFax	simple	Optional	The fax number for the address. See section 2.8.2.44 Fax for further details.
originEmail	simple	Optional	The email address for the address. See section 2.8.2.45 Email for further details.
destinationZoneName	simple	Optional	A name of the carrier zone that the receiver's address is matched to. See section 2.8.2.99 ZoneName for further details.
destinationContactName	simple	Optional	A contact name for an individual at the destination location. See section 2.8.2.61 ContactName for further details.
destinationCompanyName	simple	Optional	The company name (if relevant) for the receiver. See section 2.8.2.59 CompanyName for further details.
destinationStreet	simple	Optional	The street details for the address. See section 2.8.2.37 Address for further details.
destinationSuburb	simple	Optional	The suburb for the address. See section 2.8.2.38 Suburb for further details.
destinationState	simple	Optional	The state for the address. See section 2.8.2.39 State for further details.
destinationCode	simple	Optional	The postal code for the address. See section 2.8.2.41 PostalCode for further details.
destinationCountry	simple	Optional	The country for the address. See section 2.8.2.42 CountryCode for further details.
destinationPhone1	simple	Optional	The first phone number for the address. See section 2.8.2.43 Phone for further details.
destinationPhone2	simple	Optional	The second phone number for the address. See section 2.8.2.43 Phone for further details.
destinationFax	simple	Optional	The fax number for the address. See section 2.8.2.44 Fax for further details.

destinationEmail	simple	Optional	The email address for the address. See section 2.8.2.45 Email for further details.
------------------	--------	----------	---

Used in: 2.6.10 getManifest.

Example:

```
<dispatchDetails xsi:type="com:DispatchDetails">
  <consignmentNumber>UVW40002</consignmentNumber>
  <consignmentDate>2009-07-02</consignmentDate>
  <distanceMeasurementType>Centimetres</distanceMeasurementType>
  <weightMeasurementType>Kilograms</weightMeasurementType>
  <actualWeight>10</actualWeight>
  <chargeableWeight>8</chargeableWeight>
  <length>50</length>
  <width>50</width>
  <height>50</height>
  <currency>AUD</currency>
  <basePrice>10.00</basePrice>
  <tax>1.00</tax>
  <totalPrice>11.00</totalPrice>
  <description>bicycle parts.</description>
</dispatchDetails>
```

2.8.2. Simple Data Types

2.8.2.1. YesNoOption

```
<xsd:simpleType name="YesNoOption">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Y"/>  
    <xsd:enumeration value="N"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

If an element requires an optional condition then the option of ‘Y’ relates to Yes and ‘N’ relates to No.

Example: “Y”

2.8.2.2. Sex

```
<xsd:simpleType name="Sex">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="M"/>  
    <xsd:enumeration value="F"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

If an object has a sex then the option of ‘M’ relates to Male and ‘F’ relates to Female.

Example: “F”

2.8.2.3. Class

```
<xsd:simpleType name="Class">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Freight"/>  
    <xsd:enumeration value="General Goods"/>  
    <xsd:enumeration value="Vehicle"/>  
    <xsd:enumeration value="Boat"/>  
    <xsd:enumeration value="Animal"/>  
    <xsd:enumeration value="Refrigerated"/>  
    <xsd:enumeration value="Other"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The class of the item being sent.

Example: “Freight”

2.8.2.4. Subclass

```
<xsd:simpleType name="Subclass">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="20"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
</xsd:restriction>  
</xsd:simpleType>
```

The subclass of the item being sent.

Example: “Car”

2.8.2.5. Mode

```
<xsd:simpleType name="Mode">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Truckload"/>  
    <xsd:enumeration value="Less than load"/>  
    <xsd:enumeration value="Container"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The mode of the item being sent.

Example: “Truckload”

2.8.2.6. TISubclass

```
<xsd:simpleType name="TISubclass">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="not required"/>  
    <xsd:enumeration value="1"/>  
    <xsd:enumeration value="2"/>  
    <xsd:enumeration value="3"/>  
    <xsd:enumeration value="4"/>  
    <xsd:enumeration value="5"/>  
    <xsd:enumeration value="6"/>  
    <xsd:enumeration value="7"/>  
    <xsd:enumeration value="8"/>  
    <xsd:enumeration value="9"/>  
    <xsd:enumeration value="10"/>  
    <xsd:enumeration value="11"/>  
    <xsd:enumeration value="12"/>  
    <xsd:enumeration value="13"/>  
    <xsd:enumeration value="14"/>  
    <xsd:enumeration value="15"/>  
    <xsd:enumeration value="16"/>  
    <xsd:enumeration value="17"/>  
    <xsd:enumeration value="18"/>  
    <xsd:enumeration value="19"/>  
    <xsd:enumeration value="20"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The number of pallets being sent.

Example: “not required”

2.8.2.7. Packaging

```
<xsd:simpleType name="Packaging">  
  <xsd:restriction base="xsd:string">
```

```

<xsd:enumeration value="Backpack"/>
<xsd:enumeration value="Bale"/>
<xsd:enumeration value="Box"/>
<xsd:enumeration value="Bunch"/>
<xsd:enumeration value="Bundle"/>
<xsd:enumeration value="Carton"/>
<xsd:enumeration value="Crate"/>
<xsd:enumeration value="Cylinder"/>
<xsd:enumeration value="Document Envelope"/>
<xsd:enumeration value="Drum"/>
<xsd:enumeration value="Flat Pack"/>
<xsd:enumeration value="Letter"/>
<xsd:enumeration value="Pail"/>
<xsd:enumeration value="Pallet"/>
<xsd:enumeration value="Parcel"/>
<xsd:enumeration value="Pipe"/>
<xsd:enumeration value="Roll"/>
<xsd:enumeration value="Satchel/Bag"/>
<xsd:enumeration value="Skid"/>
<xsd:enumeration value="Suitcase"/>
<xsd:enumeration value="Tube"/>
<xsd:enumeration value="Unpackaged or N/A"/>
<xsd:enumeration value="Wheel/Tyre"/>
</xsd:restriction>
</xsd:simpleType>

```

The type of packaging used for the item being sent.

Example: “Bale”

2.8.2.8. PalletType

```

<xsd:simpleType name="PalletType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Chep"/>
    <xsd:enumeration value="Loscam"/>
    <xsd:enumeration value="Plain"/>
    <xsd:enumeration value="Not Required"/>
  </xsd:restriction>
</xsd:simpleType>

```

The pallet type being sent.

Example: “Chep”

2.8.2.9. PalletNature

```

<xsd:simpleType name="PalletNature">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Exchange"/>
    <xsd:enumeration value="Transfer"/>
    <xsd:enumeration value="Not Required"/>
  </xsd:restriction>
</xsd:simpleType>

```

The pallet nature being sent.

Example: “Exchange”

2.8.2.10. ContainerDimensions

```
<xsd:simpleType name="ContainerDimensions">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="20FT"/>  
    <xsd:enumeration value="40FT"/>  
    <xsd:enumeration value="48FT"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The dimensions of the container being sent.

Example: "20FT"

2.8.2.11. ContainerNature

```
<xsd:simpleType name="ContainerNature">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Shipping Container"/>  
    <xsd:enumeration value="Flat Rack"/>  
    <xsd:enumeration value="Open Top Container"/>  
    <xsd:enumeration value="High Cube"/>  
    <xsd:enumeration value="Curtin Sider"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The nature of the container being sent.

Example: "Shipping Container"

2.8.2.12. VehicleMake

```
<xsd:simpleType name="VehicleMake">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The make of the vehicle being sent.

Example: "Ford"

2.8.2.13. VehicleModel

```
<xsd:simpleType name="VehicleModel">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The model of the vehicle being sent.

Example: “Mustang”

2.8.2.14. VehicleDescription

```
<xsd:simpleType name="VehicleDescription">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The description of the vehicle being sent.

Example: “GT Convertible 2dr Auto 3sp 289”

2.8.2.15. VehicleYear

```
<xsd:simpleType name="VehicleYear">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1920"/>  
    <xsd:maxInclusive value="2050"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The year of the vehicle being sent.

Example: “1965”

2.8.2.16. VehicleRegistration

```
<xsd:simpleType name="VehicleRegistration">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="20"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The registration of the vehicle being sent.

Example: “123ABC”

2.8.2.17. BoatMake

```
<xsd:simpleType name="BoatMake">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The make of the boat being sent.

Example: “Haines”

2.8.2.18. BoatModel

```
<xsd:simpleType name="BoatModel">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The model of the boat being sent.

Example: "Signature 1550"

2.8.2.19. BoatHullType

```
<xsd:simpleType name="BoatHullType">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The hull type of the boat being sent.

Example: "Deep-V Hull"

2.8.2.20. LivestockType

```
<xsd:simpleType name="LivestockType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Cattle"/>  
    <xsd:enumeration value="Horse"/>  
    <xsd:enumeration value="Sheep"/>  
    <xsd:enumeration value="Pig"/>  
    <xsd:enumeration value="Other"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of the livestock being sent.

Example: "Cattle"

2.8.2.21. AnimalType

```
<xsd:simpleType name="AnimalType">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of the animal being sent.

Example: "Ardvark"

2.8.2.22. AnimalBreed

```
<xsd:simpleType name="AnimalBreed">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The breed of the animal being sent.

Example: "Fox Terrier"

2.8.2.23. AnimalAge

```
<xsd:simpleType name="AnimalAge">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The age of the animal being sent.

Example: "1"

2.8.2.24. AnimalCrate

```
<xsd:simpleType name="AnimalCrate">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Hire 1"/>  
    <xsd:enumeration value="Hire 2"/>  
    <xsd:enumeration value="Buy 1"/>  
    <xsd:enumeration value="Buy 2"/>  
    <xsd:enumeration value="I have a crate"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of crate to be used to transport the animal being sent.

Example: "I have a crate"

2.8.2.25. DistanceMeasurementType

```
<xsd:simpleType name="DistanceMeasurementType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Centimetres"/>  
    <xsd:enumeration value="Metres"/>  
    <xsd:enumeration value="Inches"/>  
    <xsd:enumeration value="Feet"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The distance measurement type to be used for the Length, Width and Height values.

Example: “Centimetres”

2.8.2.26. WeightMeasurementType

```
<xsd:simpleType name="WeightMeasurementType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Grams"/>
    <xsd:enumeration value="Kilograms"/>
    <xsd:enumeration value="Ounces"/>
    <xsd:enumeration value="Pounds"/>
  </xsd:restriction>
</xsd:simpleType>
```

The weight measurement type to be used for the Weight values.

Example: “Kilograms”

2.8.2.27. Length

```
<xsd:simpleType name="Length">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

The length of the item being sent.

Example: “50”

2.8.2.28. Width

```
<xsd:simpleType name="width">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

The width of the item being sent.

Example: “50”

2.8.2.29. Height

```
<xsd:simpleType name="Height">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

The height of the item being sent.

Example: “50”

2.8.2.30. Weight

```
<xsd:simpleType name="weight">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="99999999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The weight of the item being sent.

Example: “50”

2.8.2.31. Quantity

```
<xsd:simpleType name="Quantity">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="99999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The number of items being sent.

Example: “2”

2.8.2.32. ItemDescription

```
<xsd:simpleType name="ItemDescription">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="1000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Further details of the item being sent, that have not been specified in other elements.

Example: “Vehicle’s Carburettor extends out of bonnet by 10cm.”

2.8.2.33. Instructions

```
<xsd:simpleType name="Instructions">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="30"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Instructions to be printed on any consignment note or booking sheet that are to be provided to the carrier agent.

Note: these should only be instructions that are not available as options, eg. do not specify “unattended delivery” as the fields “destinationResUnattended” or “destinationBusUnattended” in section 2.8.1.3 Anywhere provide this information.

Example: “Please use side door.”

2.8.2.34. Comments

```
<xsd:simpleType name="Comments">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="5000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Any general comments that relate to the request, but not specific to the delivery instructions.

Example: “If you need to contact me urgently please use my phone contacts as I rarely check my email.”

2.8.2.35. DeliveryNature

```
<xsd:simpleType name="DeliveryNature">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Domestic"/>  
    <xsd:enumeration value="International"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Whether the item is being sent domestically (ie. within the same country) or internationally (from one country to another).

Example: “Domestic”

2.8.2.36. DeliveryType

```
<xsd:simpleType name="DeliveryType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Door to Door"/>  
    <xsd:enumeration value="Depot to Depot"/>  
    <xsd:enumeration value="Port to Port"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The delivery method for the item being sent.

Example: “Door to Door”

2.8.2.37. Address

```
<xsd:simpleType name="Address">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The street or postal address of a location.

Example: “1 Transport Street”

2.8.2.38. Suburb

```
<xsd:simpleType name="Suburb">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Suburbs are commonly defined as smaller residential communities lying immediately outside a city.

Further to the restrictions applied above the following rules will also be applied after the data has passed the XML tests:

- If address is in Australia: must match exactly (ie. letter for letter but not case sensitive) with the suburb name defined within the Temando system, which are extracted from the Postcode Datafile as provided by Australia Post (<https://www.auspost.com.au>).

Example: “Spring Hill”

2.8.2.39. State

```
<xsd:simpleType name="State">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="[A-Z]{3}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The 3 letter code representing a state which is a land area of local government within a country. Sometimes also referred to as a county, province or region.

Further to the restrictions applied above the following rules will also be applied after the data has passed the XML tests:

- If address is in Australia must be one of the following: QLD, NSW, VIC, ACT, SA, WA, NT or TAS.

Example: “QLD”

2.8.2.40. City

```
<xsd:simpleType name="City">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A city or town within a State or Country.

Example: "Brisbane"

2.8.2.41. PostalCode

```
<xsd:simpleType name="PostalCode">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="10"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A postal code (known in various countries as a post code, postcode, or ZIP code) is a series of letters and/or digits appended to a postal address for the purpose of sorting mail.

Further to the restrictions applied above the following rules will also be applied after the data has passed the XML tests:

- If address is in Australia: must match exactly (ie. letter for letter but not case sensitive) with the postal codes defined within the Temando system, which are extracted from the Postcode Datafile as provided by Australia Post (<https://www.auspost.com.au>).

Example: "4001"

2.8.2.42. CountryCode

```
<xsd:simpleType name="CountryCode">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="[A-Z]{2}"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The 2 letter code representing a country as specified by ISO 3166 (ie. International Standards Organisation – <http://www.iso.org>).

Example: "AU"

2.8.2.43. Phone

```
<xsd:simpleType name="Phone">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A telephone number.

Example: “(07) 3333 3333”

2.8.2.44. Fax

```
<xsd:simpleType name="Fax">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A fax number.

Example: “(07) 3333 3333”

2.8.2.45. Email

```
<xsd:simpleType name="Email">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="500"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

An e-mail address identifying a location to which e-mail messages can be delivered.

Example: “john.doe@zyxtransport.com.au”

2.8.2.46. Website

```
<xsd:simpleType name="Website">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="300"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A Uniform Resource Locator (URL) that specifies where an identified resource is available and the mechanism for retrieving it.

Example: “https://www.zyxtransport.com.au”

2.8.2.47. PortName

```
<xsd:simpleType name="PortName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

A port name used for sending or receiving items for international transportation.

Example: “Brisbane”

2.8.2.48. PortType

```
<xsd:simpleType name="PortType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Sea"/>
    <xsd:enumeration value="Air"/>
    <xsd:enumeration value="Any"/>
  </xsd:restriction>
</xsd:simpleType>
```

What type of transportation is supported for a port.

Example: “Sea”

2.8.2.49. LocationType

```
<xsd:simpleType name="LocationType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Residence"/>
    <xsd:enumeration value="Business"/>
  </xsd:restriction>
</xsd:simpleType>
```

What type of location is either sending an item or receiving it.

Example: “Residence”

2.8.2.50. Date

```
<xsd:simpleType name="Date">
  <xsd:restriction base="xsd:string">
    <xsd:pattern
      value="20[0-9]{2}-(0[1-9]|1[012])-(0[1-9]|[12][0-9]|3[01])"/>
  </xsd:restriction>
</xsd:simpleType>
```

A date specified using the international format defined by ISO 8601 (ie. International Standards Organisation – <http://www.iso.org>) which follows the numerical date system as follows: YYYY-MM-DD.

Example: for 02nd March, 2009 as “2009-03-02”

2.8.2.51. Datetime

```
<xsd:simpleType name="Datetime">
  <xsd:restriction base="xsd:string">
    <xsd:pattern
```



```

value="20[0-9]{2}-(0[1-9]|1[012])-(0[1-9]|12)[0-9]|3[01])
T([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]Z"/>
</xsd:restriction>
</xsd:simpleType>

```

A datetime specified using the international format defined by ISO 8601 (ie. International Standards Organisation – <http://www.iso.org>) which follows the numerical date system as follows: YYYY-MM-DDThh:mm:ssZ. The datetime provided will be a Coordinated Universal Time (UTC), see http://en.wikipedia.org/wiki/Coordinated_Universal_Time for further details.

Example: for 2:26pm and 12 seconds on the 02nd March, 2009 as “2009-03-02T14:26:12Z”

2.8.2.52. ReadyTime

```

<xsd:simpleType name="ReadyTime">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="AM"/>
    <xsd:enumeration value="PM"/>
  </xsd:restriction>
</xsd:simpleType>

```

A period of the day separated into 2 parts.

Example: “PM”

2.8.2.53. CurrencyAmount

```

<xsd:simpleType name="CurrencyAmount">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="8"/>
    <xsd:fractionDigits value="2"/>
    <xsd:maxInclusive value="999999.99"/>
    <xsd:minInclusive value="-999999.99"/>
  </xsd:restriction>
</xsd:simpleType>

```

The representation of a monetary amount, excluding currency symbols or country codes, as used in Australia.

Example: “123.45”

2.8.2.54. CurrencyType

```

<xsd:simpleType name="CurrencyType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="AUD"/>
  </xsd:restriction>
</xsd:simpleType>

```

The 3 letter code representing a currency code as defined within ISO 4217 (ie. International Standards Organisation – <http://www.iso.org>).

Note: currently only the Australian currency is available.

Example: “AUD”

2.8.2.55. DeliveryMethod

```
<xsd:simpleType name="DeliveryMethod">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The method used for transporting an item.

Example: “Off-peak”

2.8.2.56. Eta

```
<xsd:simpleType name="Eta">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="999999999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The number of days estimated for delivery.

Example: “2”

2.8.2.57. AdjustmentDescription

```
<xsd:simpleType name="AdjustmentDescription">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The description of an adjustment that can be or is applied to a quote amount.

Example: “Credit Payment Adjustment”

2.8.2.58. CarrierId

```
<xsd:simpleType name="CarrierId">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="999999999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The unique id for a carrier.

Example: “654321”

2.8.2.59. **CompanyName**

```
<xsd:simpleType name="CompanyName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The name of a company.

Example: “ZYX Transport”

2.8.2.60. **LocationName**

```
<xsd:simpleType name="LocationName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="20"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The description allocated to a predefined location stored in the client’s location listings.

Example: “ZYX Transport - BNE”

2.8.2.61. **ContactName**

```
<xsd:simpleType name="ContactName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The name of an individual who is the contact for a company.

Example: “John Doe”

2.8.2.62. **CarrierConditions**

```
<xsd:simpleType name="CarrierConditions">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="20000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The terms and conditions for sending an item via a carrier, as defined by that carrier (ie. not Temando's conditions but the carrier's conditions).

Example: "Our conditions include: lots of text."

2.8.2.63. DepotName

```
<xsd:simpleType name="DepotName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The name of a carrier's depot location.

Example: "BNE Office"

2.8.2.64. DepotInstructions

```
<xsd:simpleType name="DepotInstructions">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="5000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The instructions for dropping off or receiving items from a carrier's depot.

Example: "Come to the Red door."

2.8.2.65. InclusionSummary

```
<xsd:simpleType name="InclusionSummary">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The summary of the details relating to an inclusion for a quote.

Example: "Tracking available"

2.8.2.66. InclusionDetails

```
<xsd:simpleType name="InclusionDetails">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="500"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The detailed details of an inclusion for a quote.

Example: "Tracking information will be available for this quote."

2.8.2.67. ExtraSummary

```
<xsd:simpleType name="ExtraSummary">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The summary of the details relating to an extra available for a quote.

Example: "Insurance"

2.8.2.68. ExtraDetails

```
<xsd:simpleType name="ExtraDetails">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="500"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The detailed details of an extra available for a quote.

Example: "Insurance cover up to \$5,000."

2.8.2.69. PaymentType

```
<xsd:simpleType name="PaymentType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Credit Card"/>  
    <xsd:enumeration value="Account"/>  
    <xsd:enumeration value="Credit"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The payment method to be used for paying for a monetary amount.

Example: "Credit Card"

2.8.2.70. CreditCardType

```
<xsd:simpleType name="CreditCardType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="MasterCard"/>  
    <xsd:enumeration value="visa"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of credit card being supplied, for credit card payments.

Example: "Credit Card"

2.8.2.71. CreditCardExpiryDate

```
<xsd:simpleType name="CreditCardExpiryDate">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="(0[1-9]|1[012])-20[0-9]{2}" />  
  </xsd:restriction>  
</xsd:simpleType>
```

The expiry date of the credit card being supplied, for credit card payments.

Example: "05-2010"

2.8.2.72. CreditCardNumber

```
<xsd:simpleType name="CreditCardNumber">  
  <xsd:restriction base="xsd:string">  
    <xsd:pattern value="[0-9]{16}" />  
  </xsd:restriction>  
</xsd:simpleType>
```

The number of the credit card being supplied, for credit card payments.

Example: "5163190000000000"

2.8.2.73. CreditCardName

```
<xsd:simpleType name="CreditCardName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50" />  
  </xsd:restriction>  
</xsd:simpleType>
```

The name of the owner of the credit card (as specified on the card) being supplied, for credit card payments.

Example: "JANE DOE"

2.8.2.74. PromotionCode

```
<xsd:simpleType name="PromotionCode">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="10" />  
  </xsd:restriction>  
</xsd:simpleType>
```

Temando code used for promotions or identifying sales representatives.

Example: "E0001"

2.8.2.75. BookingNumber

```
<xsd:simpleType name="BookingNumber">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The booking number allocated by the carrier (associated with a quote) representing that the request has been confirmed and booked.

Example: "XYZ300001"

2.8.2.76. ConsignmentNumber

```
<xsd:simpleType name="ConsignmentNumber">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The carrier's consignment number identifies a logical grouping of goods that has been consigned.

Example: "UVW400001"

2.8.2.77. ConsignmentDocument

```
<xsd:simpleType name="ConsignmentDocument">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="1000000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The carrier's consignment document.

Important: The contents of this field are encoded using base64.

This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.

When creating a file from this field's contents you will need to ensure that you decode using base64.

Example: "JVBER ... *shortened for readability* ... XRhbG"

2.8.2.78. ConsignmentDocumentType

```
<xsd:simpleType name="ConsignmentDocumentType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="application/pdf"/>  
    <xsd:enumeration value="application/msword"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

```
<xsd:enumeration value="application/excel"/>
</xsd:restriction>
</xsd:simpleType>
```

The type of consignment document specified in the corresponding field of type 2.8.2.77 ConsignmentDocument.

These types are the standard Multipurpose Internet Mail Extensions (MIME) as defined by the Internet Engineering Task Force (IETF) (please refer to <http://www.mimetypes.org/>).

The document types supported include:

- application/pdf – Adobe PDF document (extension .pdf)
- application/msword - Microsoft Word document (extension .doc)
- application/excel – Microsoft Excel spreadsheet (extension .xls)

Example: “application/pdf”

2.8.2.79. ManifestDocument

```
<xsd:simpleType name="ManifestDocument">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="1000000"/>
  </xsd:restriction>
</xsd:simpleType>
```

The carrier's manifest document.

Important: The contents of this field are encoded using base64.

This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.

When creating a file from this field's contents you will need to ensure that you decode using base64.

Example: “JVBER ... *shortened for readability* ... XRhbG”

2.8.2.80. ManifestDocumentType

```
<xsd:simpleType name="ManifestDocumentType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="application/pdf"/>
    <xsd:enumeration value="application/msword"/>
    <xsd:enumeration value="application/excel"/>
  </xsd:restriction>
</xsd:simpleType>
```

The type of consignment document specified in the corresponding field of type 2.8.2.79 ManifestDocument.

These types are the standard Multipurpose Internet Mail Extensions (MIME) as defined by the Internet Engineering Task Force (IETF) (please refer to <http://www.mimetypes.org/>).

The document types supported include:

- application/pdf – Adobe PDF document (extension .pdf)
- application/msword - Microsoft Word document (extension .doc)
- application/excel – Microsoft Excel spreadsheet (extension .xls)

Example: “application/pdf”

2.8.2.81. LabelDocument

```
<xsd:simpleType name="LabelDocument">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="1000000"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The carrier's label document.

Important: The contents of this field are encoded using base64.

This is done to ensure that the binary data that forms the file's contents survive transport through the transport layer.

When creating a file from this field's contents you will need to ensure that you decode using base64.

Example: “e3Rf3 ... shortened for readability ... X8ggy”

2.8.2.82. LabelDocumentType

```
<xsd:simpleType name="LabelDocumentType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="application/pdf"/>  
    <xsd:enumeration value="application/msword"/>  
    <xsd:enumeration value="application/excel"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of label document specified in the corresponding field of type 2.8.2.81 LabelDocument.

These types are the standard Multipurpose Internet Mail Extensions (MIME) as defined by the Internet Engineering Task Force (IETF) (please refer to <http://www.mimetypes.org/>).

The document types supported include:

- application/pdf – Adobe PDF document (extension .pdf)
- application/msword - Microsoft Word document (extension .doc)

application/excel – Microsoft Excel spreadsheet (extension .xls)

Example: “application/pdf”

2.8.2.83. TrackingStatus

```
<xsd:simpleType name="TrackingStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Unavailable"/>
    <xsd:enumeration value="Issue"/>
    <xsd:enumeration value="Awaiting Pickup"/>
    <xsd:enumeration value="In Transit"/>
    <xsd:enumeration value="Delivered"/>
  </xsd:restriction>
</xsd:simpleType>
```

The status of the transit of the item/s being sent.

Can be represented as:

Unavailable –	tracking information is not available for this carrier.
Issue –	an issue has occurred in the transit of the item/s.
Awaiting Pickup –	item/s have not been picked up from the origin location.
In Transit –	item/s are currently in transit (ie. have been picked up but not yet delivered).
Delivered –	item/s have been delivered to the destination location.

Example: “Delivered”

2.8.2.84. TrackingFurtherDetails

```
<xsd:simpleType name="TrackingFurtherDetails">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="300"/>
  </xsd:restriction>
</xsd:simpleType>
```

Further details on the transit of the item/s being sent.

Example: “Nobody was present at the destination location. Request submitted did not have the unattended option selected, therefore person was required to receive freight. Item currently at the Brisbane depot. Contact carrier for further information.”

2.8.2.85. RequestStatus

```
<xsd:simpleType name="RequestStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Awaiting Acceptance"/>
    <xsd:enumeration value="Awaiting Booking"/>
    <xsd:enumeration value="Completed"/>
    <xsd:enumeration value="Cancelled"/>
  </xsd:restriction>
</xsd:simpleType>
```

The status of an existing request within the Temando system.

Can be represented as:

- Awaiting Acceptance – quote has been accepted. Awaiting request confirmation and payment.
- Awaiting Booking – quote has been accepted, request confirmed and payment received. Awaiting a booking reference number to be supplied by the carrier (confirming the booking with the carrier).
- Completed – quote has been accepted, request confirmed, payment received and booking reference assigned by carrier.
- Cancelled – request has been cancelled. No further action will be taken for this request.

Example: “Completed”

2.8.2.86. GeneratedType

```
<xsd:simpleType name="GeneratedType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Manually"/>
    <xsd:enumeration value="Automatically"/>
  </xsd:restriction>
</xsd:simpleType>
```

How a quote was generated, either automatically or manually.

Can be represented as:

- Automatically – price created based on the carrier’s pre-recorded rates.
- Manually – the carrier has reviewed the request and then provided a price.

Example: “Automatically”

2.8.2.87. LoginId

```
<xsd:simpleType name="LoginId">
  <xsd:restriction base="xsd:string">
```

```
<xsd:maxLength value="50"/>
</xsd:restriction>
</xsd:simpleType>
```

Login id for an account.

Example: "jsmith"

2.8.2.88. Password

```
<xsd:simpleType name="Password">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="20"/>
  </xsd:restriction>
</xsd:simpleType>
```

Password for an account.

Example: "my_password1357"

2.8.2.89. IndividualCompany

```
<xsd:simpleType name="IndividualCompany">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Individual"/>
    <xsd:enumeration value="Company"/>
  </xsd:restriction>
</xsd:simpleType>
```

Whether a client account is for an individual or a proprietary company.

Can be represented as:

Individual – individual person.

Company – proprietary company.

Example: "Individual"

2.8.2.90. Surname

```
<xsd:simpleType name="Surname">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="50"/>
  </xsd:restriction>
</xsd:simpleType>
```

An individual's surname.

Example: "Smith"

2.8.2.91. Firstname

```
<xsd:simpleType name="Firstname">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

An individual's first name.

Example: "John"

2.8.2.92. CompanyNumber

```
<xsd:simpleType name="CompanyNumber">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="30"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The number issued by the governing body (either at a state or federal level) responsible for the registration and renewal of a company's details. For example in Australia the number issued by ASIC is an ABN (Australian Business Number).

Example: "64 123 456 789"

2.8.2.93. ClientId

```
<xsd:simpleType name="ClientId">  
  <xsd:restriction base="xsd:int">  
    <xsd:minInclusive value="1"/>  
    <xsd:maxInclusive value="999999999"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The unique id for a client.

Example: "123456"

2.8.2.94. CarrierReference

```
<xsd:simpleType name="CarrierReference">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The reference code used by a carrier company to refer to a particular request.

Example: "XYZ100000"

2.8.2.95. CarrierAccountNumber

```
<xsd:simpleType name="CarrierAccountNumber">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="20"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The reference code used by a carrier company to refer to the Temando account.

Example: "TEM00001"

2.8.2.96. CarrierCreatorId

```
<xsd:simpleType name="CarrierCreatorId">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="40"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The reference code used by a carrier company to refer to a third party (ie. not Temando) that created a request.

Example: "NOTTEM001"

2.8.2.97. RateName

```
<xsd:simpleType name="RateName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="50"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The rate classification for an item or a group of items, used for rates calculation.

Example: "Carton"

2.8.2.98. Cubic

```
<xsd:simpleType name="Cubic">  
  <xsd:restriction base="xsd:decimal">  
    <xsd:totalDigits value="8"/>  
    <xsd:fractionDigits value="4"/>  
    <xsd:maxInclusive value="9999.9999"/>  
    <xsd:minInclusive value="0"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The representation of a cubic value, which is the result of the multiplication of the length, width and height of an object. Example, an object of length 0.5m, width 0.2m and height 0.1m would have a cubic value of 0.01m³ (ie. 0.5 x 0.2 x 0.1).

Example: "0.01"

2.8.2.99. ZoneName

```
<xsd:simpleType name="ZoneName">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The name representing a zonal region, used for rates calculation.

Example: "X123456"

2.8.2.100. LocationPosition

```
<xsd:simpleType name="LocationPosition">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Origin"/>  
    <xsd:enumeration value="Destination"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Whether a location is an originating address (ie. pickup) or a destination address (dropoff).

Can be represented as:

Origin – a originating address (ie. pickup).

Destination – a destination address (ie. dropoff).

Example: "Origin"

2.8.2.101. ManifestType

```
<xsd:simpleType name="ManifestType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Awaiting Confirmation"/>  
    <xsd:enumeration value="Confirmed"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of manifest report to be generated.

Can be represented as:

Awaiting Confirmation – those requests that have not yet been confirmed.

Confirmed – those requests that have been confirmed.

Example: “Awaiting Confirmation”

2.8.2.102. ClientReference

```
<xsd:simpleType name="ClientReference">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="100"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The reference code used by a client to refer to a particular request.

Example: “ABC200000”

2.8.2.103. LabelPrinterType

```
<xsd:simpleType name="LabelPrinterType">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Thermal"/>  
    <xsd:enumeration value="Standard"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The type of printer that will be used to print the label.

Can be represented as:

Thermal – for thermal label printers.

Standard – general laser printers.

Example: “Standard”

2.8.2.104. Detail

```
<xsd:simpleType name="Detail">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="Detailed"/>  
    <xsd:enumeration value="Summary"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

The amount of information to be returned.

Can be represented as:

Detailed – all available information.

Summary – limited amount of information.

Example: “Detailed”

3. Appendix A: soapUI Support

soapUI is a well known tool for web service testing. A freeware version (including open source) is available for download from the internet.

With this tool you will be able to replicate the calls and responses that you will send and receive when using the Temando API from a code base.

The soapUI tool is available from <http://www.soapui.org/>.

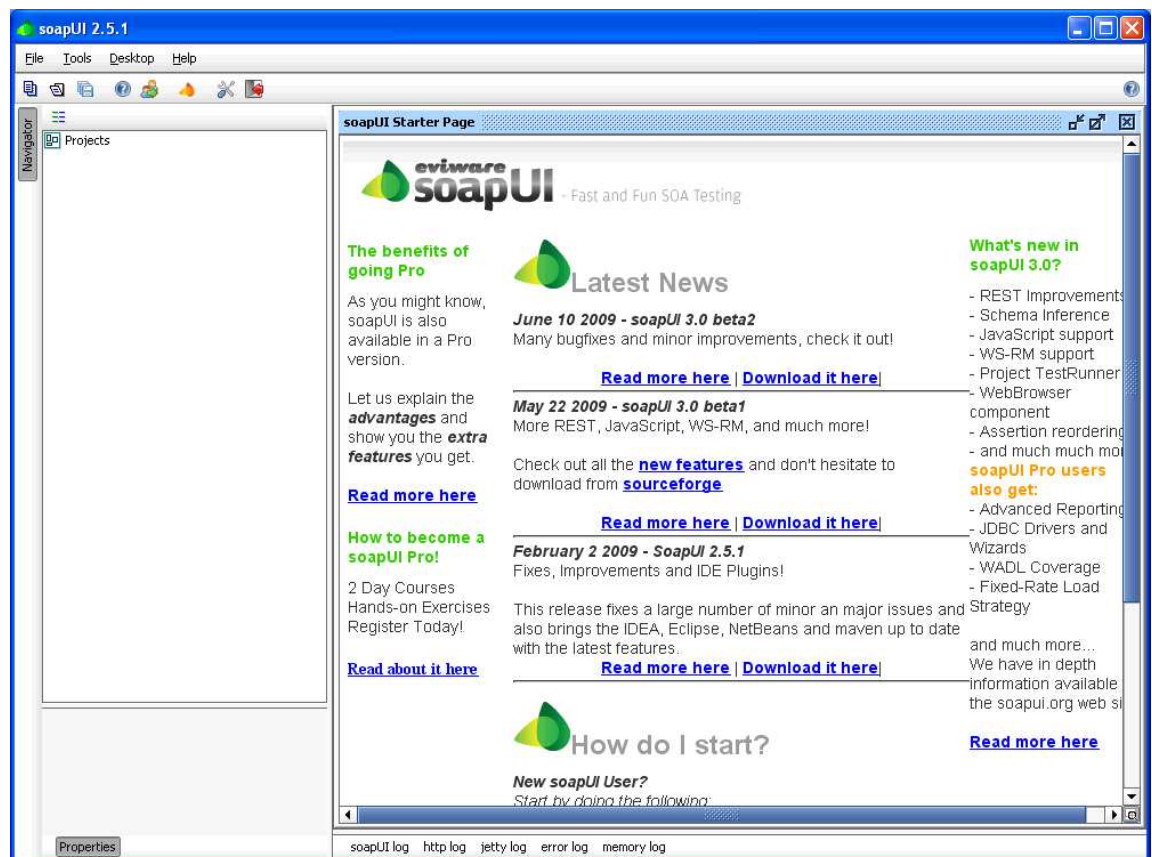
3.1. Setup

Step 1: Download and install soapUI

Download and install the soapUI application. Instructions for this process are available from the soapUI website (see above).

Step 2: Run the application

Run the application. The soapUI window will be displayed.



Step 3: Create the project/s

Create a new project by right clicking on “Projects” in the main navigator section (left area of the window) and selecting the New soapUI project. The New soapUI Project dialog box will be displayed.



There are 2 web services available to users of the API. The first is the live service which contains the live data for the Temando system. The second is the training service which contains fictitious data that can be used for testing.

The following are the values to enter for both the live and training services:

LIVE

Project Name: Temando Live

Initial WSDL/WADL: https://api.temando.com/schema/2009_06/server.wsdl

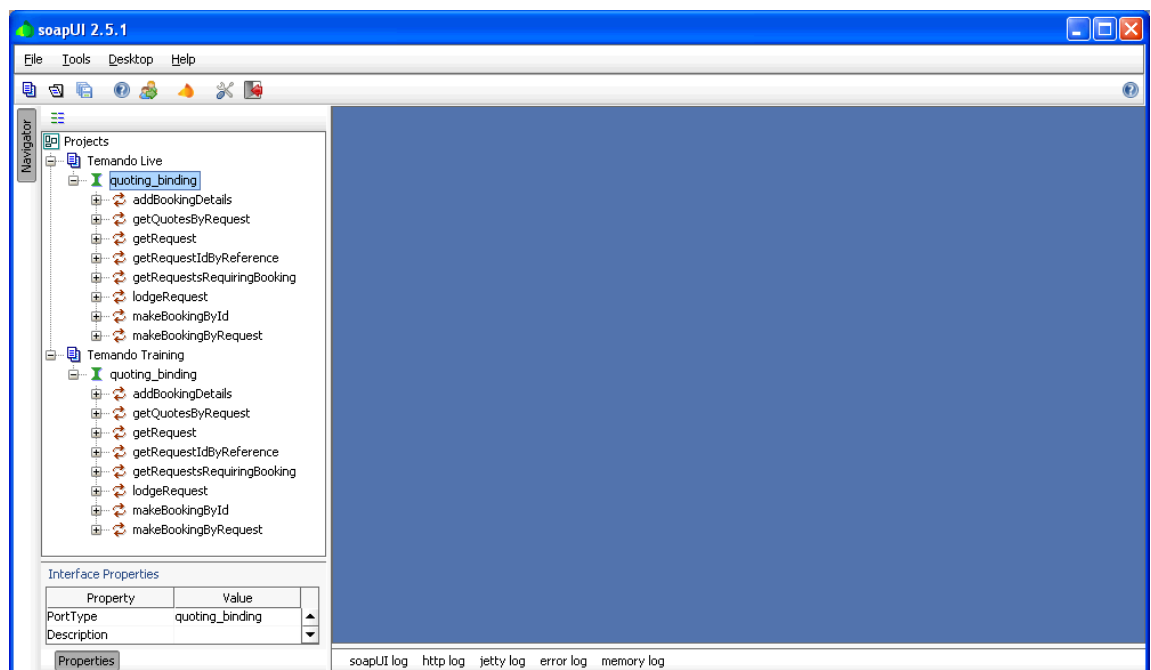
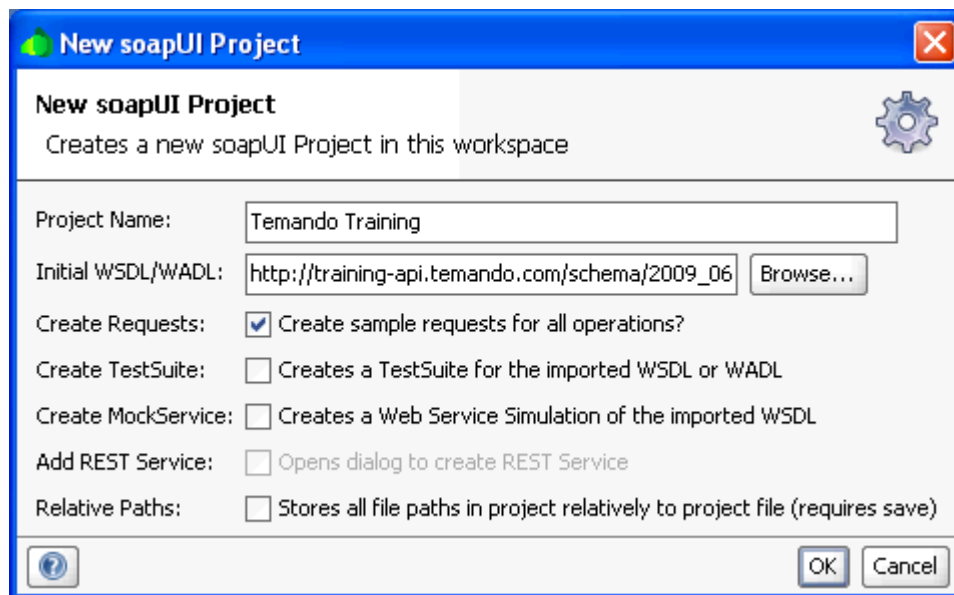
TRAINING

Project Name: Temando Training

Initial WSDL/WADL: https://training-api.temando.com/schema/2009_06/server.wsdl

Once you have entered either the live or training values into the New soapUI Project dialog box click on the Ok button.

Note: You are not limited to creating projects for just one service, a project can be created for both services. Just repeat the process in this step for each service.



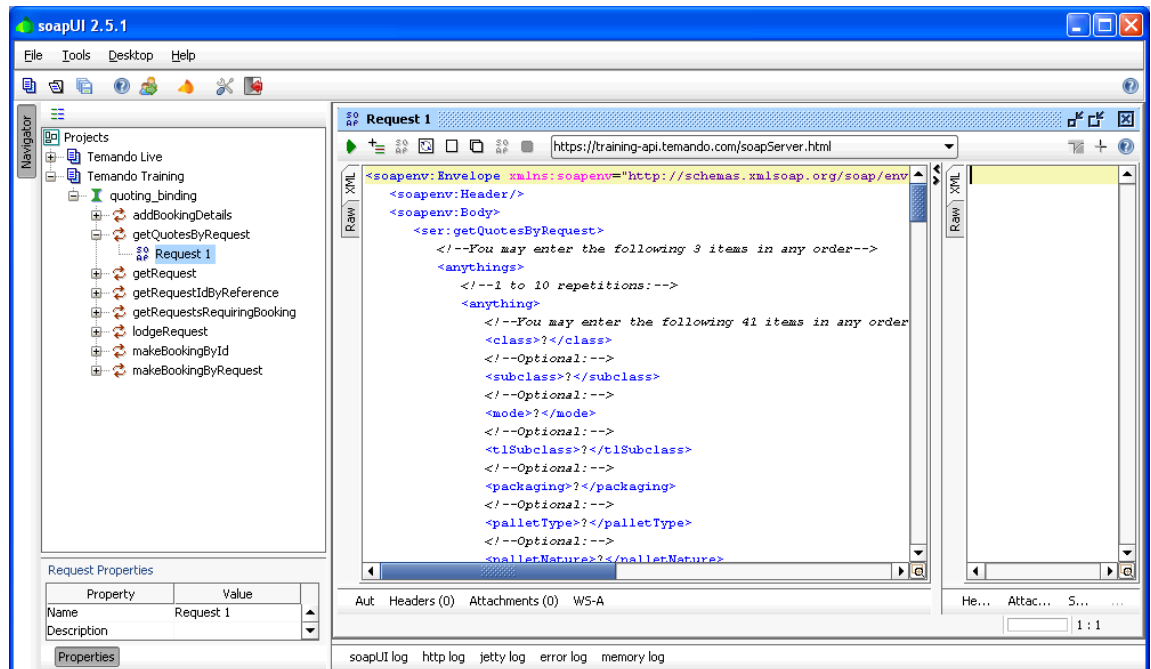
3.2. Example Call

For this example a call will be made to the training service calling the `getQuotesByRequest` method.

You will need to have followed the steps in section 3.1 Setup before being able to perform this example.

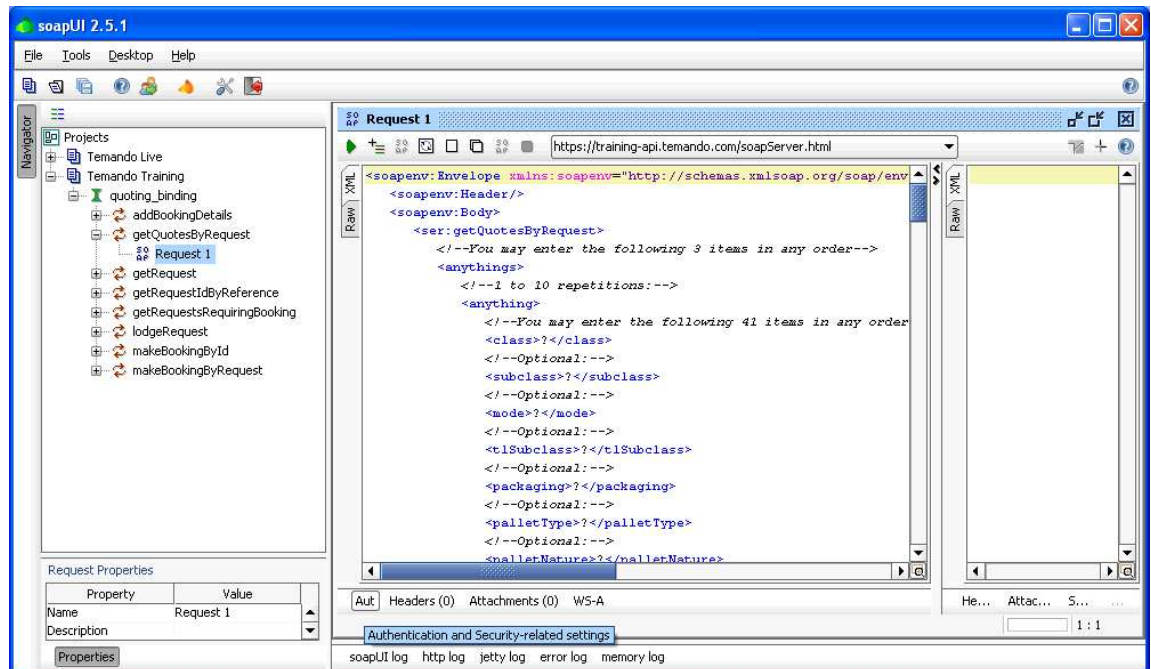
Step 1: Open a new request

Expand out the tree for the path *Temando Training* → *quoting_binding* → *getQuotesByRequest* → *Request 1*, located in the main navigator section (left area of the window). Double click “Request 1”. The Request 1 window will be displayed in the soapUI desktop section (right area of the window).



Step 2: Enter login details

Click on “Auth” located in the bottom left of soapUI desktop section (bottom right area of the window).



The following are the values to enter for both the live and training services:

LIVE

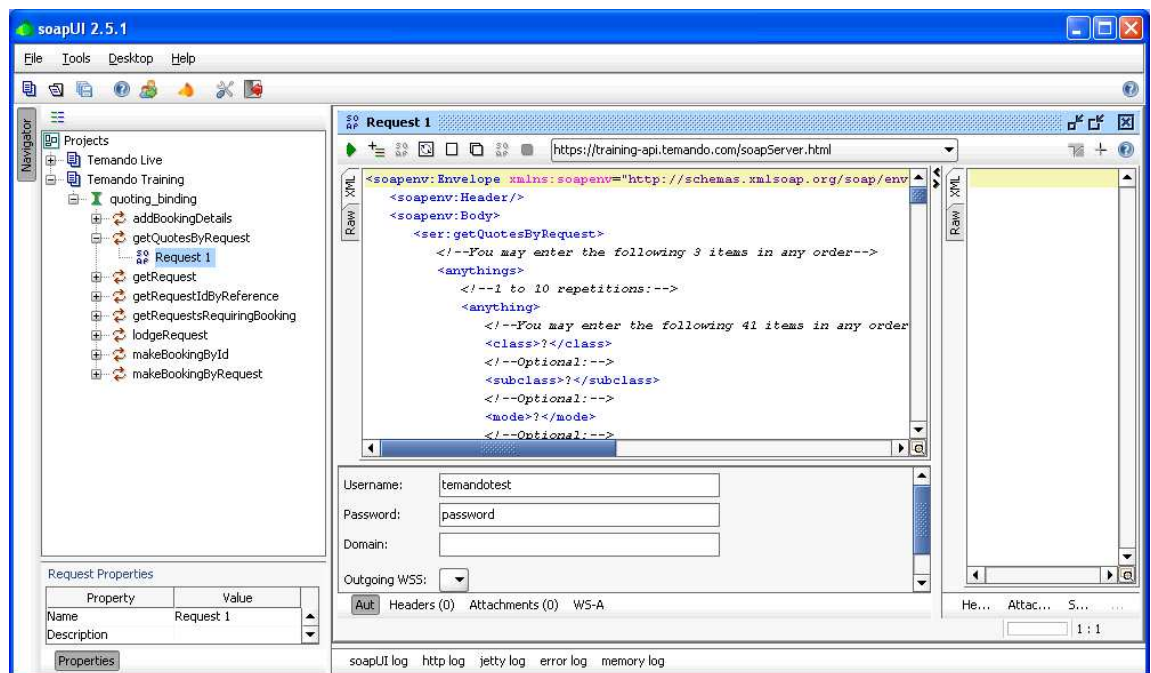
Username: <your login>
Password: <your password>

TRAINING (CLIENT)

Username: temandotest
Password: password

TRAINING (AFFILIATE)

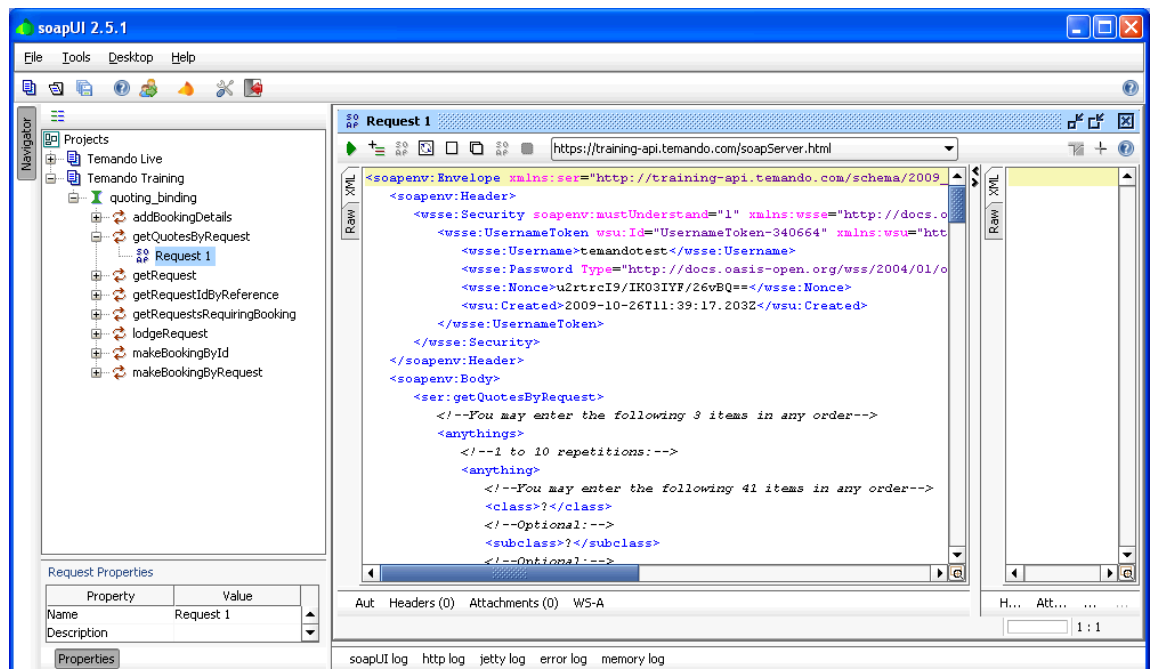
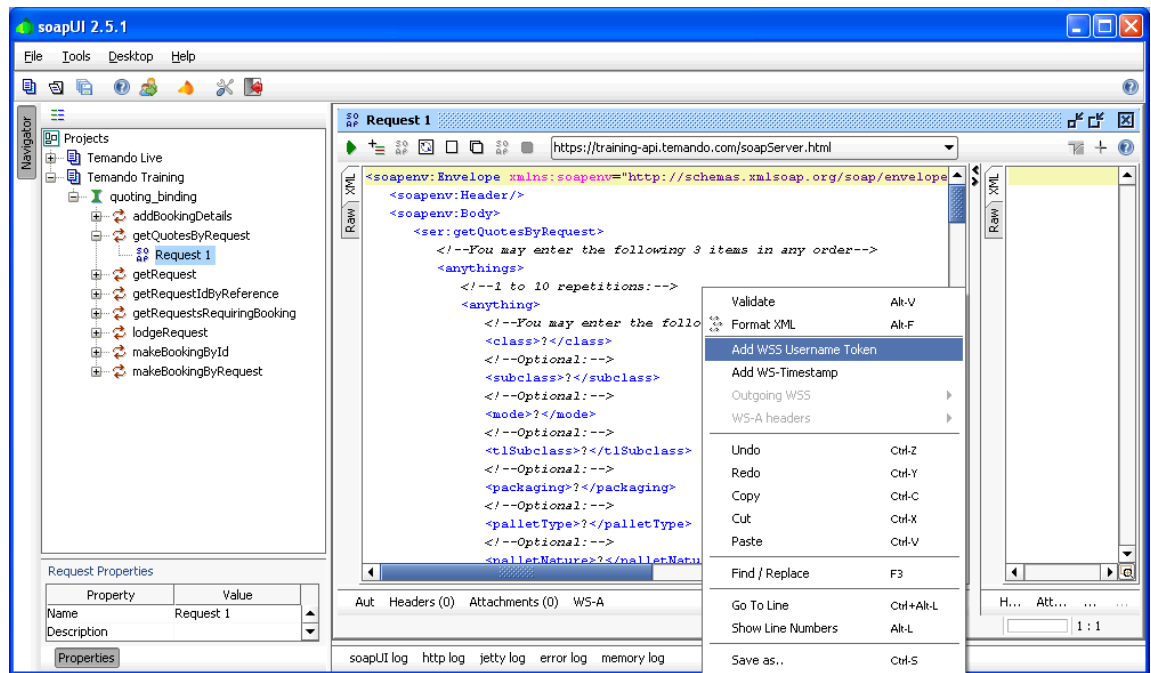
Username: affiliatet
Password: password



Once you have entered the login values you can hide this section by clicking on “Auth” again.

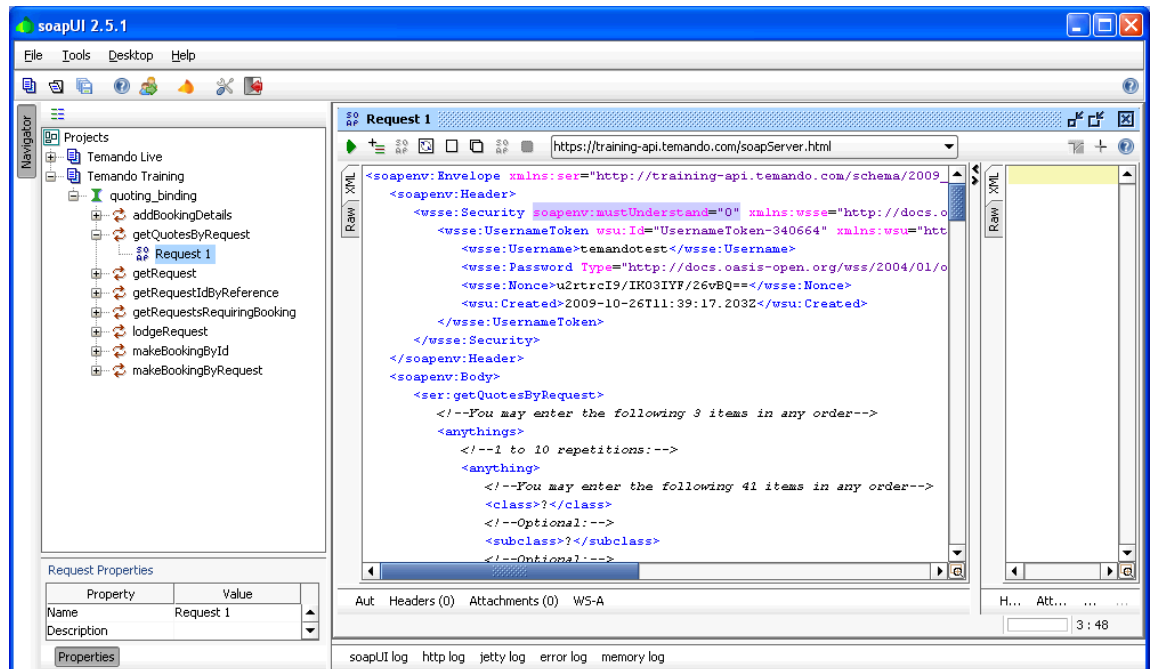
Step 3: Include login details in request

Right click anywhere inside the new request and select the option Add WSS Username Token. When prompted with the Specify Password Type dialog select the option PasswordText and click on Ok. The username token is now embedded in the request.



Important: In the Security tag make sure you disable the mustUnderstand element (ie. change the assigned 1 to 0). See the below screen shot if unable to locate this element.

This check is not required as security checks will be performed at the server side.



Step 4: Enter the request details

Update the contents of the Body tag so that the XML is exactly the same as the following:

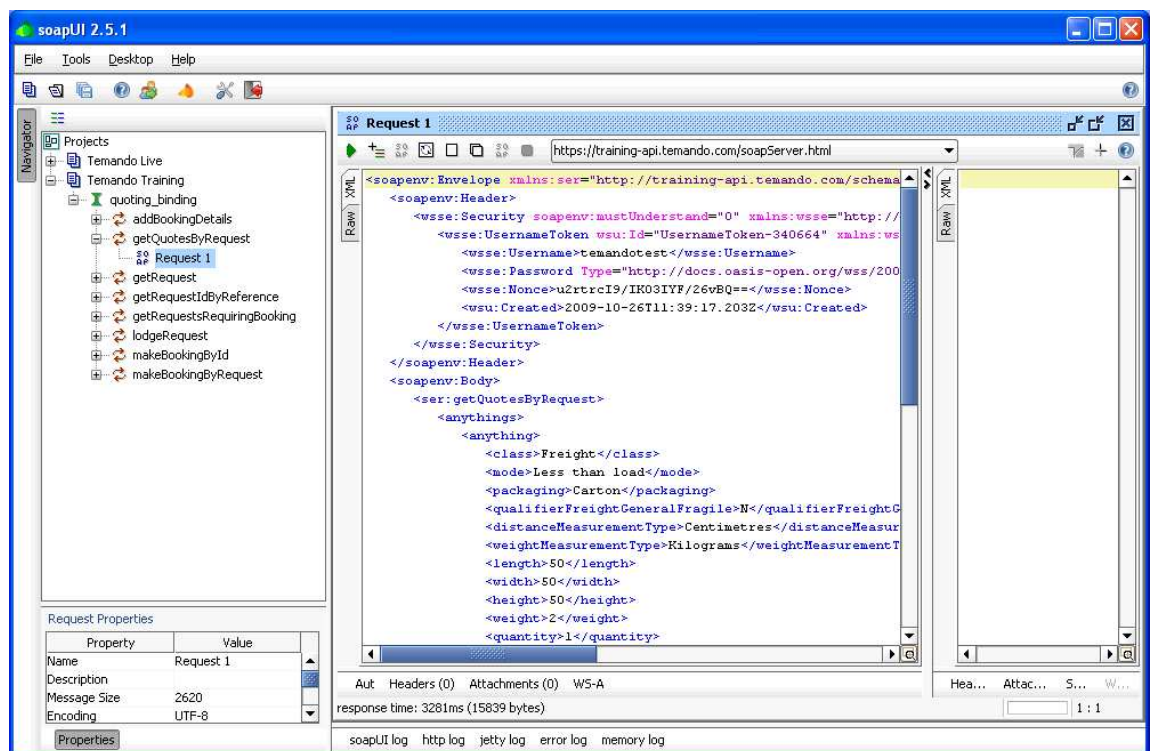
```
<soapenv:Body>
  <ser:getQuotesByRequest>
    <anythings>
      <anything>
        <class>Freight</class>
        <mode>Less than load</mode>
        <packaging>Carton</packaging>
        <qualifierFreightGeneralFragile>N</qualifierFreightGeneralFragile>
        <distanceMeasurementType>Centimetres</distanceMeasurementType>
        <weightMeasurementType>Kilograms</weightMeasurementType>
        <length>50</length>
        <width>50</width>
        <height>50</height>
        <weight>2</weight>
        <quantity>1</quantity>
        <description>Contains bottle lids.</description>
      </anything>
    </anythings>
  </ser:getQuotesByRequest>
</soapenv:Body>
```



```

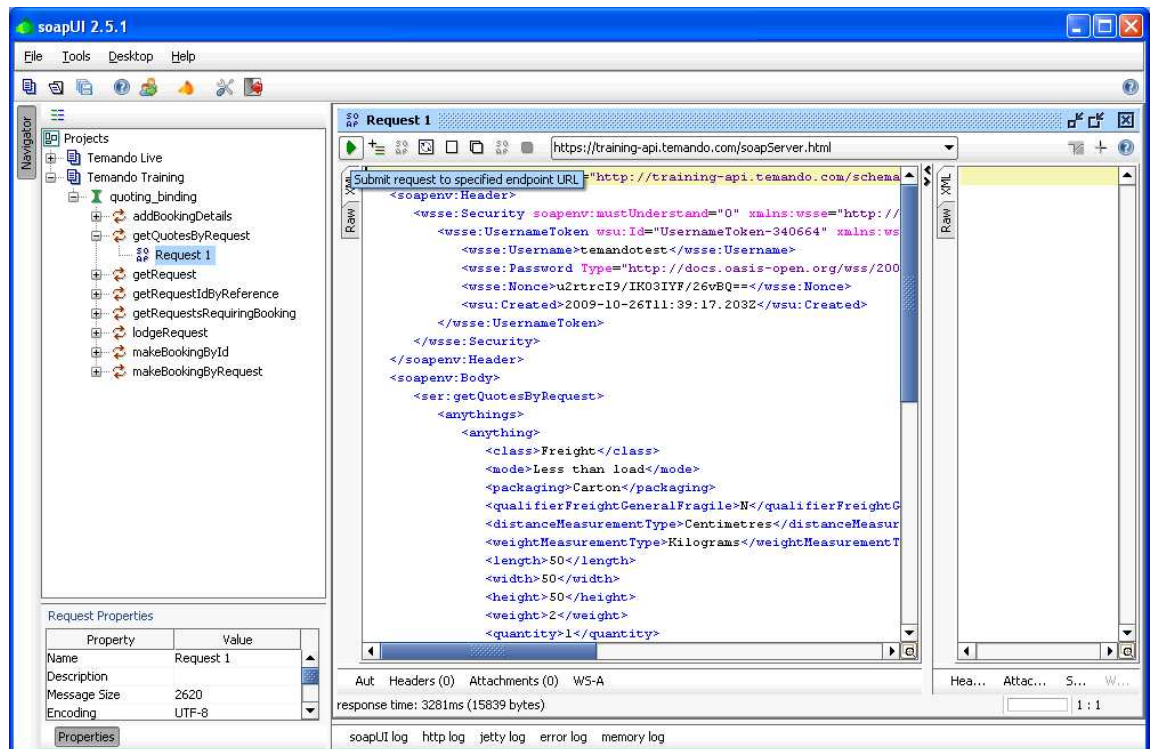
<originIs>Business</originIs>
<originBusUnattended>N</originBusUnattended>
<originBusDock>Y</originBusDock>
<originBusForklift>N</originBusForklift>
<originBusLoadingFacilities>N</originBusLoadingFacilities>
<originBusInside>N</originBusInside>
<originBusLimitedAccess>N</originBusLimitedAccess>
<originBusHeavyLift>N</originBusHeavyLift>
<originBusTailgateLifter>N</originBusTailgateLifter>
<originBusContainersSwingLifter>N</originBusContainersSwingLifter>
</anywhere>
<anytime>
  <readyDate>2009-12-01</readyDate>
  <readyTime>PM</readyTime>
</anytime>
<general>
  <goodsValue>2000.00</goodsValue>
</general>
<instructions>Please use side door.</instructions>
</ser:getQuotesByRequest>
</soapenv:Body>

```

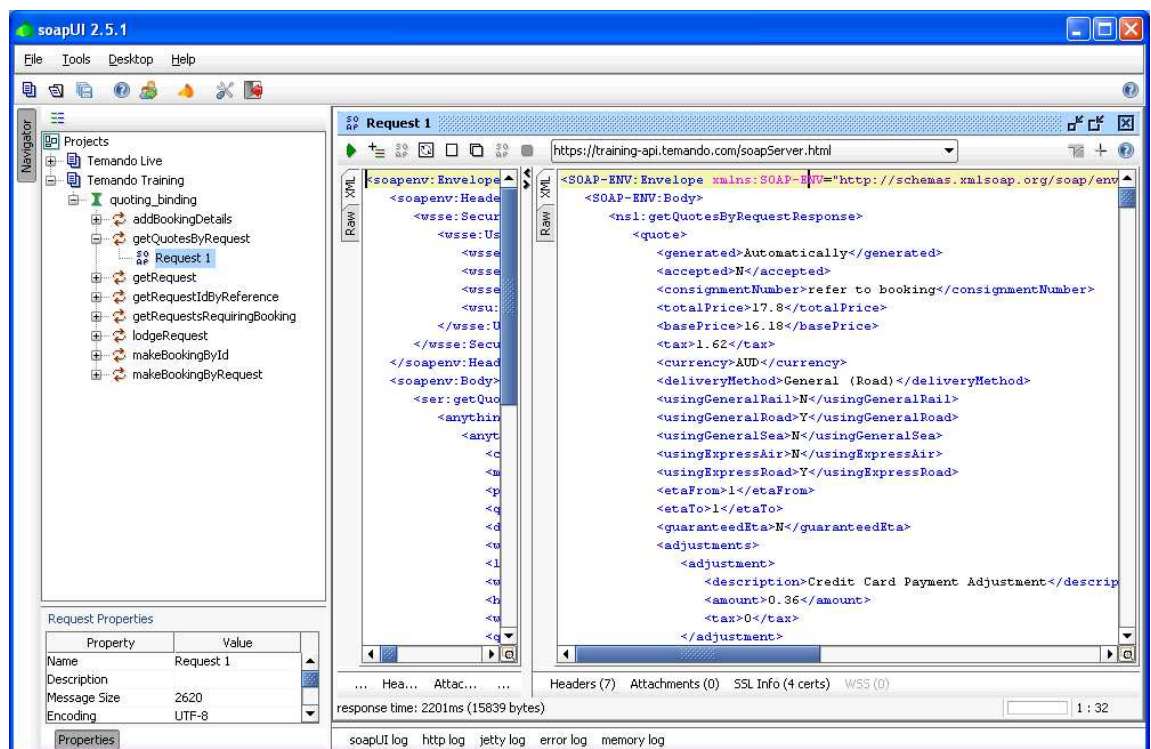


Step 5: Call the web service

Click on the green arrow located in the top left area of the soapUI desktop section (right area of the window). This will send the request to the web service.



The results of the call will be returned to the right area of the soapUI desktop section (right area of the window).



4. Appendix B: PHP Example

4.1. Overview

The following section provides example PHP code for calling the `getQuotesByRequest` method from the Training web service.

The code includes examples of:

- Creating a `SoapClient`
- Setting header information
- Structuring request information
- Calling a method
- Extracting the values from the response.

4.2. SOAP Installation Testing

Note: before you can use this example in your own development environment you will need to ensure that you have the SOAP implementation included. A simple way to test for this is to create a new soap test file on your server that displays the results of the `phpinfo()` call. The code is as follows:

```
<?php
phpinfo();
?>
```

Run the soap test page in a browser and search for the word “SOAP”. If there is a section entitled “soap” then you do have the SOAP libraries installed. If not, you will need to install them. Details on this installation process are located at <http://php.net/manual/en/book.soap.php>.

Important: If you have created a soap test file (detailed above) for this SOAP test make sure that you delete this file after you have finished testing as end-users would be able to view all your PHP system settings if they load it.

4.3. Code

```
<?php
```

```

////////////////////////////////////
//
// Details:      This file contains example PHP code to be used as a guide for
//               setting up a web service connection and call to the Temando
//               service.
//               This example connects to the Training web service using the
//               WSDL file located at:
//               https://training-api.temando.com/schema/2009_06/server.wsdl
//               The call used for this example is the getQuotesByRequest call
//               located in section 2.5.1. getQuotesByRequest of the reference
//               documentation.
// Created:      2009-09-21.
// Created By:   Temando Pty Ltd.
// Updated:      no updates since creation.
// References:   Temando API R1_01 Developers Guide.doc
// Copyright:    Copyright 2009 Temando Pty Limited. All rights reserved.
//               Not to be copied, redistributed or modified without prior
//               written consent of Temando Pty Limited.
//
////////////////////////////////////

//////////////////////////////////// CREATE SOAPCLIENT AND ASSIGN HEADER INFORMATION //////////////////////////////////////
// Create and configure the SoapClient, adding the security header information to the
// client.

// The WSDL cache should be set to on to prevent the WSDL being loaded everytime.
ini_set("soap.wsdl_cache_enabled", "1");

// Create a new SoapClient referencing the Temando WSDL file.
$client = new SoapClient("https://training-api.temando.com/schema/2009_06/server.wsdl",
    array('soap_version' => SOAP_1_2));

// Define the username (ie. login id) and password for connecting to the Temando
// web service. These are the same setting used for your website access.
$username = "temandotest";
$password = "password";

// Define the security string that wraps your login details. Due to limitations
// with the PHP language this header information can only be provided via a string.
$headerSecurityStr
    = "<Security><UsernameToken><Username>".$username."</Username><Password>".$password."</Pass
    word></UsernameToken></Security>";

// Create a new SoapVar using the header security string.
$headerSecurityVar = new SoapVar($headerSecurityStr, XSD_ANYXML);

// Create a new SoapHeader containing all your login details.
$soapHeader = new SoapHeader('wsse:http://schemas.xmlsoap.org/ws/2002/04/secext',
    'soapenv:Header', $headerSecurityVar);

// Add the SoapHeader to your SoapClient.
$client->__setSoapHeaders( array($soapHeader) );

////////////////////////////////////

//////////////////////////////////// STRUCTURE REQUEST INFORMATION //////////////////////////////////////
// Create objects for containing the details of the request to be send.

// Array for containing a list of all the Anything entries.
$anythings = array();

// Array for containing the first (and only) Anything details.
// Refer to section 2.7.1.2. Anything (of the reference documentation) for further
// details on element types.
$anything1 = array();
$anything1["class"] = "Freight";
$anything1["mode"] = "Less than load";
$anything1["packaging"] = "Carton";
$anything1["qualifierFreightGeneralFragile"] = "N";
$anything1["distanceMeasurementType"] = "Centimetres";
$anything1["weightMeasurementType"] = "Kilograms";
$anything1["length"] = "50";
$anything1["width"] = "50";
$anything1["height"] = "50";
$anything1["weight"] = "2";
$anything1["quantity"] = "1";
$anything1["description"] = "Contains bottle lids.";

```

```

array_push($anythings, $anything1);

// Array for containing the Anywhere details.
// Refer to section 2.7.1.3. Anywhere (of the reference documentation) for further
// details on element types.
$anywhere = array();
$anywhere["itemNature"] = "Domestic";
$anywhere["itemMethod"] = "Door to Door";
$anywhere["originCountry"] = "AU";
$anywhere["originCode"] = "4000";
$anywhere["originSuburb"] = "Brisbane";
$anywhere["destinationCountry"] = "AU";
$anywhere["destinationCode"] = "2000";
$anywhere["destinationSuburb"] = "Sydney";
$anywhere["destinationIs"] = "Residence";
$anywhere["destinationResPostalBox"] = "N";
$anywhere["destinationResUnattended"] = "N";
$anywhere["destinationResInside"] = "N";
$anywhere["destinationResLimitedAccess"] = "N";
$anywhere["destinationResHeavyLift"] = "N";
$anywhere["originIs"] = "Business";
$anywhere["originBusUnattended"] = "N";
$anywhere["originBusDock"] = "Y";
$anywhere["originBusForklift"] = "N";
$anywhere["originBusLoadingFacilities"] = "N";
$anywhere["originBusInside"] = "N";
$anywhere["originBusLimitedAccess"] = "N";
$anywhere["originBusHeavyLift"] = "N";
$anywhere["originBusTailgateLifter"] = "N";
$anywhere["originBusContainerSwingLifter"] = "N";

// Array for containing the Anytime details.
// Refer to section 2.7.1.4. Anytime (of the reference documentation) for further
// details on element types.
$anytime = array();
$anytime["readyDate"] = "2009-12-01";
$anytime["readyTime"] = "PM";

// Array for containing the General details.
// Refer to section 2.7.1.5. General (of the reference documentation) for further
// details on element types.
$general = array();
$general["goodsValue"] = "2000.00";

// Variable for containing the Instructions details.
// Refer to section 2.7.2.33. Instructions (of the reference documentation) for further
// details on element types.
$instructions = "Please use side door.";

// Array for containing the Request details. This includes the Anythings, Anywhere
// and Anytime details created above.
// Refer to section 2.5.1.2 Request - getQuotesByRequest (of the reference documentation)
// for further details on element types.
$getQuotesByRequest = array();
$getQuotesByRequest["anythings"] = $anythings;
$getQuotesByRequest["anytime"] = $anytime;
$getQuotesByRequest["anywhere"] = $anywhere;
$getQuotesByRequest["general"] = $general;
$getQuotesByRequest["instructions"] = $instructions;

////////////////////////////////////

//////////////////////////////////// CALL METHOD //////////////////////////////////////
// Call the method getQuotesByRequest using the request details defined above.

// Call the method within a try-catch block to ensure any errors generated from
// the method call are captured as an exception.
try{

    // Call the method using the request details.
    $getQuotesByRequestResponse = $client->getQuotesByRequest($getQuotesByRequest);

} catch (SoapFault $exception){

    // Nothing is done with the exception yet. Will be processed later in the display
    code.

```

```

}

////////////////////////////////////

?>

<!-- ----- DISPLAY RESULTS OF CALL ----- -->
<!-- Display the results of the call in HTML format. -->

<!-- Page display header information. -->
<html>
<body bgcolor="#4A4A4A">
<center>
<div id="API_example" align=center style="position:relative; height:400px; width:750px;
padding:5px; border: #000000 1px solid; background-color:#FFFFFF; visibility: visible;">
<br>
<p style="color: #F78421; font-weight: bold; text-decoration: underline; font-size:
18px;">TEST API CALL</p>

<?php

//////////////////////////////////// EXCEPTION THROWN DISPLAY //////////////////////////////////////
// If an exception was thrown from the call display the following information.

if( isset($exception) ){

// Exception thrown from the Temando web service contain 2 pieces of information.
// The first is the faultcode which is a Temando defined code for identifying the
// error generated. The second is the faultstring which provides human-readable
// details on the exception thrown.
// All faults are defined within section 2.6 SOAP Faults (of the reference
// documentation).

?>

        <div id="API_error" align=left style="position:relative; height:200px;
width:700px; padding:5px; border: 0px; background-color:#E1F0FF; visibility: visible;">

            <p style="color: black; font-size: 16px; font-weight: bold; text-decoration:
underline;">Error</p>

            <p style="color: black; font-size: 12px;">The specific details of the error
generated are as follows:</p>

            <table>
            <tr>
            <th width="200"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Name</p></th>
            <th width="400"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Contents</p></th>
            </tr>

<?php

            // Display the faultcode.
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>faultcode</p></td><td><p style=\"color: #848484; font-size: 12px;\>\".$exception-
>faultcode.\"</p></td></tr>";
            // Display the faultstring.
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>faultstring</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$exception->faultstring.\"</p></td></tr>";

?>

        </table>

        </div>

<?php

////////////////////////////////////

```

```

////////// SUCCESSFUL CALL DISPLAY //////////
// If no exception was thrown and a response was received from the call, display
// the following information.

} else{

?>

    <!-- Display the details of the request sent to the web service. -->

    <div id="API_request" align=left style="position:relative; height:400px;
width:700px; padding:5px; border: 0px; background-color:#E1F0FF; visibility: visible;">

        <p style="color: black; font-size: 16px; font-weight: bold; text-decoration:
underline;">Parameters Sent</p>

        <p style="color: black; font-size: 14px; font-weight: bold;">Anything (1):</p>

        <table>
        <tr>
        <th width="200"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Name</p></th>
        <th width="400"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Contents</p></th>
        </tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;"><class</p></td><td><p style="color: #848484; font-size:
12px;">Freight</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">mode</p></td><td><p style="color: #848484; font-size: 12px;">Less than
load</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">packaging</p></td><td><p style="color: #848484; font-size:
12px;">Carton</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">qualifierFreightGeneralFragile</p></td><td><p style="color: #848484; font-size:
12px;">N</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">distanceMeasurementType</p></td><td><p style="color: #848484; font-size:
12px;">Centimetres</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">weightMeasurementType</p></td><td><p style="color: #848484; font-size:
12px;">Kilograms</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">length</p></td><td><p style="color: #848484; font-size: 12px;">50</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">width</p></td><td><p style="color: #848484; font-size: 12px;">50</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">height</p></td><td><p style="color: #848484; font-size: 12px;">50</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">weight</p></td><td><p style="color: #848484; font-size: 12px;">2</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">quantity</p></td><td><p style="color: #848484; font-size: 12px;">1</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">description</p></td><td><p style="color: #848484; font-size: 12px;">Contains
bottle lids.</p></td></tr>
        </table>

        <p style="color: black; font-size: 14px; font-weight: bold;">Anywhere:</p>

        <table>
        <tr>
        <th width="200"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Name</p></th>
        <th width="400"><p style="color: black; font-size: 12px; font-weight:
bold;">Element Contents</p></th>
        </tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">itemNature</p></td><td><p style="color: #848484; font-size:
12px;">Domestic</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">itemMethod</p></td><td><p style="color: #848484; font-size: 12px;">Door to
Door</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight:
bold;">originCountry</p></td><td><p style="color: #848484; font-size:
12px;">AU</p></td></tr>

```

```

        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">originCode</p></td><td><p style="color: #848484; font-size: 12px;">4000</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">originSuburb</p></td><td><p style="color: #848484; font-size: 12px;">Brisbane</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationCountry</p></td><td><p style="color: #848484; font-size: 12px;">AU</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationCode</p></td><td><p style="color: #848484; font-size: 12px;">2000</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationSuburb</p></td><td><p style="color: #848484; font-size: 12px;">Sydney</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationIs</p></td><td><p style="color: #848484; font-size: 12px;">Business</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationBusNotifyBefore</p></td><td><p style="color: #848484; font-size: 12px;">N</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">destinationBusLimitedAccess</p></td><td><p style="color: #848484; font-size: 12px;">N</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">originIs</p></td><td><p style="color: #848484; font-size: 12px;">Business</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">originBusNotifyBefore</p></td><td><p style="color: #848484; font-size: 12px;">Y</p></td></tr>
        <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">originBusLimitedAccess</p></td><td><p style="color: #848484; font-size: 12px;">N</p></td></tr>
    </table>

```

```

    <p style="color: black; font-size: 14px; font-weight: bold;">Anytime:</p>

```

```

    <table>
    <tr>
    <th width="200"><p style="color: black; font-size: 12px; font-weight: bold;">Element Name</p></th>
    <th width="400"><p style="color: black; font-size: 12px; font-weight: bold;">Element Contents</p></th>
    </tr>
    <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">readyDate</p></td><td><p style="color: #848484; font-size: 12px;">2009-12-01</p></td></tr>
    <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">readyTime</p></td><td><p style="color: #848484; font-size: 12px;">PM</p></td></tr>
    </table>

```

```

    <p style="color: black; font-size: 14px; font-weight: bold;">Instructions:</p>

```

```

    <table>
    <tr>
    <th width="200"><p style="color: black; font-size: 12px; font-weight: bold;">Element Name</p></th>
    <th width="400"><p style="color: black; font-size: 12px; font-weight: bold;">Element Contents</p></th>
    </tr>
    <tr><td><p style="color: #848484; font-size: 12px; font-weight: bold;">instructions</p></td><td><p style="color: #848484; font-size: 12px;">Please use side door.</p></td></tr>
    </table>

```

```

</div>

```

```

<br>

```

```

    <!-- Display the details of the response details received from the web service. ->

```

```

    <div id="API_response" align=left style="position:relative; height:400px; width:700px; padding:5px; border: 0px; background-color:#DFFDE3; visibility: visible;">

```

```

        <p style="color: black; font-size: 16px; font-weight: bold; text-decoration: underline;">Values Returned</p>

```

```

<?php

```



```

        // Extract the array of quotes received in the response object.
        // Refer to section 2.5.1.3 Response - getQuotesByRequest (of the reference
documentation)
        // for further details on element types.
        $quotes = $getQuotesByRequestResponse->quote;

        foreach( $quotes as $quoteKey => $quoteDetails ){

            echo "<hr><p style=\"color: black; font-size: 14px; font-weight:
bold;\>Quote (\".$quoteKey.\"):</p>";

?>

            <!-- Display the details of the current quote. -->

            <table>
            <tr>
            <th width="200"><p style="color: black; font-size: 12px; font-weight:
bold;\>Element Name</p></th>
            <th width="400"><p style="color: black; font-size: 12px; font-weight:
bold;\>Element Contents</p></th>
            </tr>

            <!-- Display quote specific information such as pricing, tax, etc. -->

            <tr>
            <td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;\>Quote Details</p></td>
            </tr>

<?php

            // Refer to section 2.7.1.4. AvailableQuote (of the reference
documentation)
            // for further details on element types.

            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>totalPrice</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$number_format($quoteDetails->totalPrice, 2).\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>basePrice</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$number_format($quoteDetails->basePrice, 2).\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>tax</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$number_format($quoteDetails->tax, 2).\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>currency</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$quoteDetails->currency.\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>deliveryMethod</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$quoteDetails->deliveryMethod.\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>etaFrom</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$quoteDetails->etaFrom.\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>etaTo</p></td><td><p style=\"color: #848484; font-size: 12px;\>\".$quoteDetails-
>etaTo.\"</p></td></tr>";
            echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>guaranteedEta</p></td><td><p style=\"color: #848484; font-size:
12px;\>\".$quoteDetails->guaranteedEta.\"</p></td></tr>";

?>

            <!-- Display adjustments associated with the quote. -->

            <tr>
            <td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;\>Adjustments</p></td>
            </tr>

<?php

            // Extract the array of adjustments associated with the current quote
object.
            // Refer to section 2.7.1.5. Adjustments (of the reference documentation)
            // for further details on element types.

            $adjustments = (array)$quoteDetails->adjustments->adjustment;

```

```

        if( sizeof($adjustments) == 0 ){
            echo "<tr><td colspan='2'><p style='color: #848484; font-size:
12px; font-style: italic;'>no adjustments.</p></td></tr>";
        } else{
            foreach( $adjustments as $adjustmentKey => $adjustmentDetails ){
                echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Adjustment      (".$adjustmentKey.") :      description</p></td><td><p
style='color: #848484; font-size: 12px;'>". $adjustmentDetails-
>description."</p></td></tr>";
                echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Adjustment      (".$adjustmentKey.") :      amount</p></td><td><p
style='color: #848484; font-size: 12px;'>$.number_format($adjustmentDetails->amount,
2)."</p></td></tr>";
                echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Adjustment (".$adjustmentKey.") : tax</p></td><td><p style='color:
#848484; font-size: 12px;'>$.number_format($adjustmentDetails->tax,
2)."</p></td></tr>";
            }
        }
    }

?>

<!-- Display inclusions associated with the quote. -->

<tr>
<td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;">Inclusions</p></td>
</tr>

<?php

// Extract the array of inclusions associated with the current quote
object.
// Refer to section 2.7.1.8. Inclusion (of the reference documentation)
// for further details on element types.

$inclusions = (array)$quoteDetails->inclusions->inclusion;

if( sizeof($inclusions) == 0 ){
    echo "<tr><td colspan='2'><p style='color: #848484; font-size:
12px; font-style: italic;'>no inclusions.</p></td></tr>";
} else{
    foreach( $inclusions as $inclusionKey => $inclusionDetails ){
        echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Inclusion (".$inclusionKey.") : summary</p></td><td><p style='color:
#848484; font-size: 12px;'>". $inclusionDetails->summary."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Inclusion (".$inclusionKey.") : details</p></td><td><p style='color:
#848484; font-size: 12px;'>". $inclusionDetails->details."</p></td></tr>";
    }
}

?>

<!-- Display extras associated with the quote. -->

<tr>
<td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;">Extras</p></td>
</tr>

<?php

// Extract the array of extras associated with the current quote object.
// Refer to section 2.7.1.11. Extra (of the reference documentation)
// for further details on element types.

$extras = (array)$quoteDetails->extras->extra;

if( sizeof($extras) == 0 ){
    echo "<tr><td colspan='2'><p style='color: #848484; font-size:
12px; font-style: italic;'>no extras.</p></td></tr>";
} else{
    foreach( $extras as $extraKey => $extraDetails ){
        echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Extra      (".$extraKey.") :      summary</p></td><td><p style='color:
#848484; font-size: 12px;'>". $extraDetails->summary."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px;
font-weight: bold;'>Extra      (".$extraKey.") :      details</p></td><td><p style='color:
#848484; font-size: 12px;'>". $extraDetails->details."</p></td></tr>";
    }
}

```

```

        echo "<tr><td><p style=\"color: #848484; font-size: 12px;
font-weight: bold;\">Extra (\".$extraKey.\"): total price</p></td><td><p style=\"color:
#848484; font-size: 12px;\">\".$extraDetails->totalPrice.\"</p></td></tr>";
    }
}

?>

<!-- Display carrier details for the quote. -->

<tr>
<td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;">Carrier Information</p></td>
</tr>

<?php

    // Extract the array containing carrier information for the current quote
    object.

    // Refer to section 2.7.1.6. Carrier (of the reference documentation)
    // for further details on element types.

    $carrier = $quoteDetails->carrier;

    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">id</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>id.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">companyName</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>companyName.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">companyContact</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->companyContact.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetAddress</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->streetAddress.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetSuburb</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->streetSuburb.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetCity</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>streetCity.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetState</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>streetState.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetCode</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>streetCode.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">streetCountry</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->streetCountry.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalAddress</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->postalAddress.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalSuburb</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->postalSuburb.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalCity</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>postalCity.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalState</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>postalState.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalCode</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>postalCode.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">postalCountry</p></td><td><p style=\"color: #848484; font-size:
12px;\">\".$carrier->postalCountry.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">phone1</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>phone1.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">phone2</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>phone2.\"</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\">email</p></td><td><p style=\"color: #848484; font-size: 12px;\">\".$carrier-
>email.\"</p></td></tr>";

```

```

        echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>website</p></td><td><p style=\"color: #848484; font-size: 12px;\>". $carrier-
>website."</p></td></tr>";
        echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-weight:
bold;\>conditions</p></td><td><p style=\"color: #848484; font-size: 12px;\>". $carrier-
>conditions."</p></td></tr>";

?>

<!-- Display originating depot details for the quote. -->

<tr>
<td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;">Originating Depot Information</p></td>
</tr>

<?php

// Extract the array containing originating depot information for the
current quote object.
// Refer to section 2.7.1.7. Depot (of the reference documentation)
// for further details on element types.

if( !isset($quoteDetails->originatingDepot) ){
    echo "<tr><td colspan=\"2\"><p style=\"color: #848484; font-size:
12px; font-style: italic;\>no depot, item being sent door-to-door.</p></td></tr>";
} else{
    $originatingDepot = $quoteDetails->originatingDepot;

    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>name</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->name."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>street</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->street."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>suburb</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->suburb."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>city</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->city."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>state</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->state."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>code</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->code."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>country</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->country."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>phone1</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->phone1."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>phone2</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->phone2."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>fax</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->fax."</p></td></tr>";
    echo "<tr><td><p style=\"color: #848484; font-size: 12px; font-
weight: bold;\>instructions</p></td><td><p style=\"color: #848484; font-size:
12px;\>". $originatingDepot->instructions."</p></td></tr>";
}

?>

<!-- Display destination depot details for the quote. -->

<tr>
<td colspan="2"><p style="color: grey; font-size: 12px; font-weight:
bold;">Destination Depot Information</p></td>
</tr>

<?php

// Extract the array containing destination depot information for the
current quote object.
// Refer to section 2.7.1.7. Depot (of the reference documentation)
// for further details on element types.

if( !isset($quoteDetails->destinationDepot) ){

```

```

        echo "<tr><td colspan='2'><p style='color: #848484; font-size:
12px; font-style: italic;'>no depot, item being sent door-to-door.</p></td></tr>";
    } else{
        $destinationDepot = $quoteDetails->destinationDepot;

        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>name</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->name."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>street</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->street."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>suburb</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->suburb."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>city</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->city."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>state</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->state."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>code</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->code."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>country</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->country."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>phone1</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->phone1."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>phone2</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->phone2."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>fax</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->fax."</p></td></tr>";
        echo "<tr><td><p style='color: #848484; font-size: 12px; font-
weight: bold;'>instructions</p></td><td><p style='color: #848484; font-size:
12px;'>". $destinationDepot->instructions."</p></td></tr>";
    }

?>

</table>

<?php
    }

?>

</div>

<?php
}

?>

<br>

</div>

</center>
</body>
</html>

<!-- ----- -->

```

4.4. Screenshots

4.4.1. Success



http://www.temando.com/temando_training3344/testAPI.html - Microsoft Internet Explorer provided by Virgin Blue

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Media Mail Print Print Preview

Address http://www.temando.com/temando_training3344/testAPI.html Go Links

Values Returned

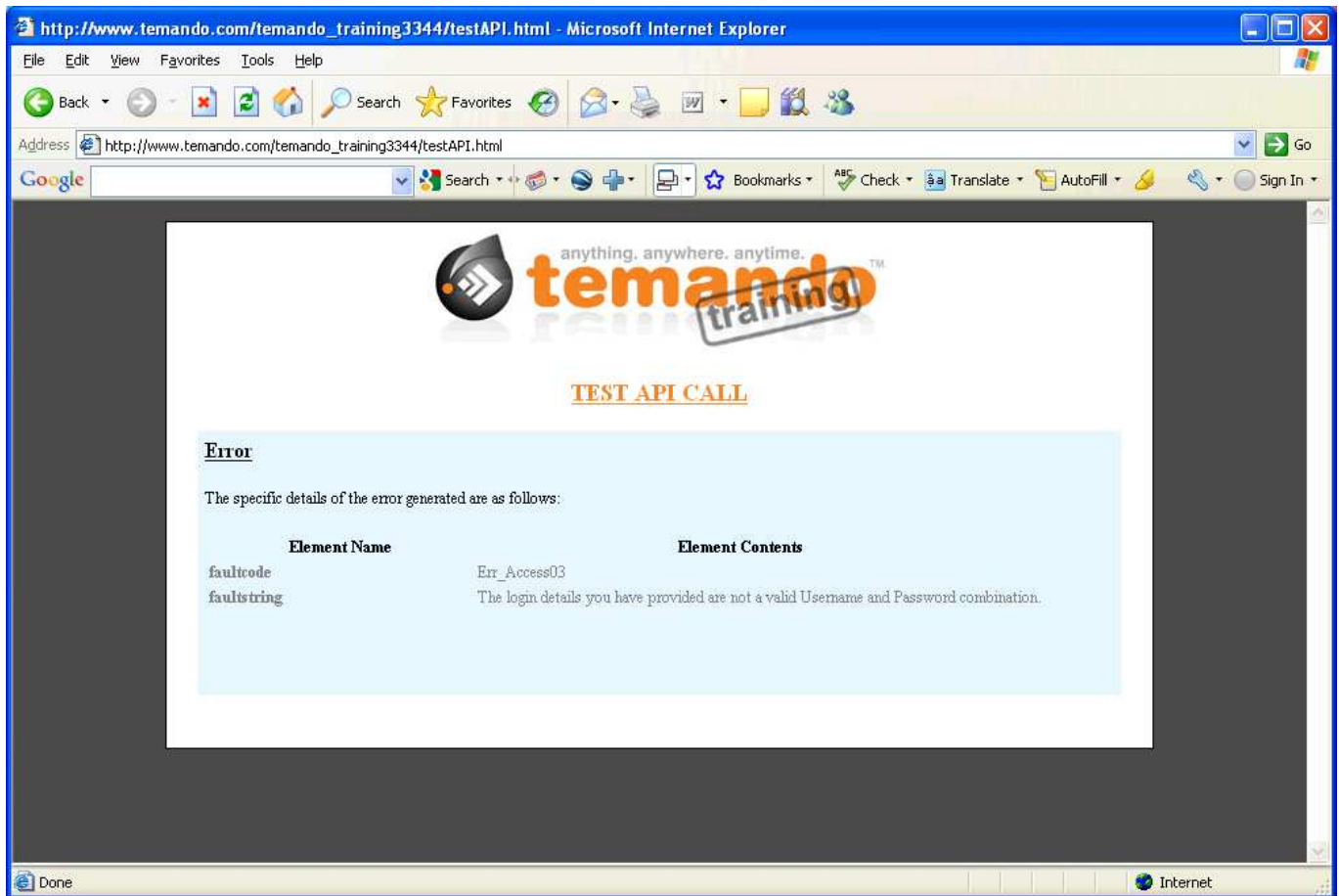
Quote (0):

Element Name	Element Contents
Quote Details	
totalPrice	\$30.61
basePrice	\$27.83
tax	\$2.78
currency	AUD
deliveryMethod	Road (Express)
etaFrom	1
etaTo	1
guaranteedEta	N
Adjustments	
Adjustment (0): description	Credit Card Payment Adjustment
Adjustment (0): amount	\$0.61
Adjustment (0): tax	\$0.00
Adjustment (1): description	Credit Payment Adjustment
Adjustment (1): amount	-\$0.56
Adjustment (1): tax	-\$0.06
Inclusions	
no inclusions.	
Carrier Information	
companyName	Hunter Express
companyContact	Stuart Poppleton
streetAddress	U7 26-32 Cosgrove Rd

Done Internet

4.4.2. Error (exception)

This example error from the API call was generated by changing the password to an incorrect value.



5. Appendix C: .NET Example

5.1. Overview

The following section provides example .NET setup procedures and code for connecting to the Temando API.

5.2. Code

This code is available by contacting your Temando contact.

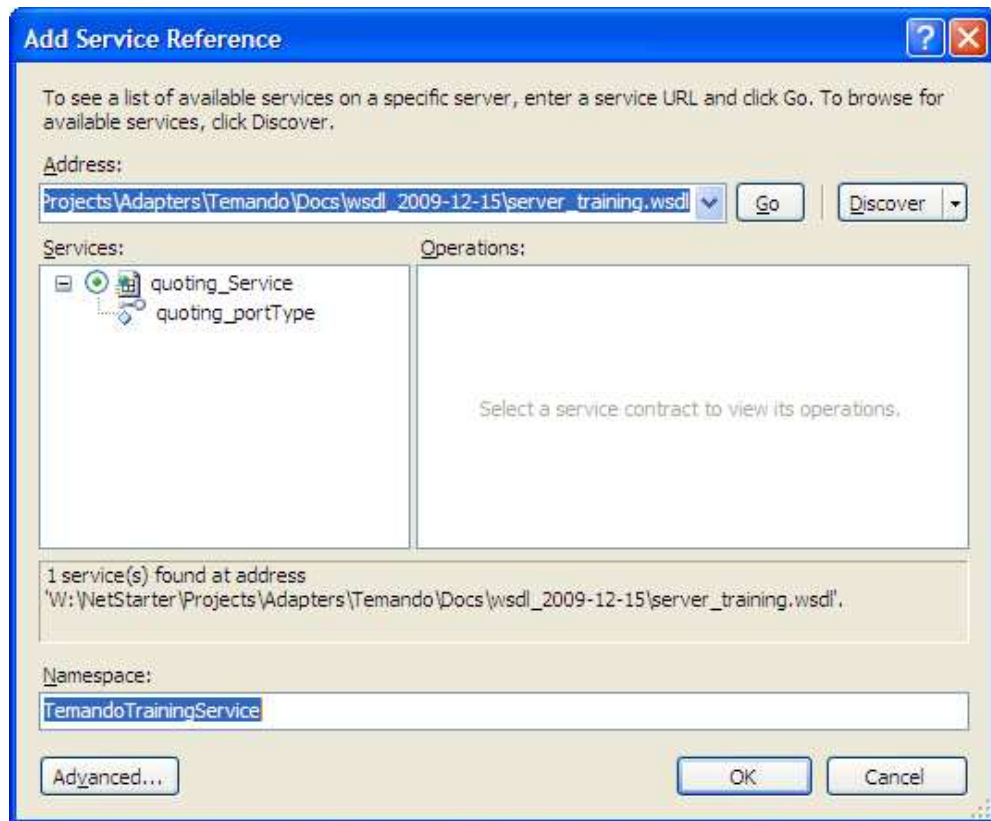
The files available include (but not limited to):

- TemandoAdapter.cs
- TemandoEndpointBehavior.cs
- TemandoLiveAdapter.cs
- TemandoMessageHeader.cs
- TemandoMessageInspector.cs
- TemandoTrainingAdapter.cs

5.3. Setup

Step 1.

Add Service References using WCF



Step 2.

Add 6 cs files to project, these files are wrappers for temando service, and they handle soap header to pass-through authentication of service.

- TemandoAdapter.cs
- TemandoEndpointBehavior.cs
- TemandoLiveAdapter.cs
- TemandoMessageHeader.cs
- TemandoMessageInspector.cs
- TemandoTrainingAdapter.cs

Step 3.

Use wrapper to call the service method.

```
` Adapter assignment for connecting to the Temando Training
API.
var adapter = new TemandoTrainingAdapter("http://training-
api.temando.com/soapServer.html", "temandotest", "password");
` Adapter assignment for connecting to the Temando Live API.
` var adapter = new
TemandoTrainingAdapter("http://api.temando.com/soapServer.htm
l", "your login", "your password");
```

```
quoting_portTypeClient client = adapter.Client;
.....
.....
AvailableQuote[] resultQuotes = client.getQuotesByRequest(
    new getQuotesByRequest
    {
        anythings = anythingsTest,
        anywhere = anywhereTest,
        anytime = anytimeTest
    });
```

For more detail, please see the code samples.

5.4. Known .NET Issues

5.4.1. Adapter Defect – Cannot Resolve/Import

This is a known .NET defect which will generate an error message as follows:

```
The document was understood, but it could not be processed.
- The WSDL document contains links that could not be resolved.
- The 'import' element is not supported in this context.
Metadata contains a reference that cannot be resolved:
'https://training-
api.temando.com/schema/2009_06/server.wsdl'.
```

This error message will be displayed if you directly reference the server.wsdl file from either the https://training-api.temando.com/schema/2009_06/server.wsdl or https://api.temando.com/schema/2009_06/server.wsdl locations.

This is due to the .NET Microsoft defect (<http://support.microsoft.com/kb/820122>) with a status of

"Microsoft has confirmed that this is a bug in the Microsoft products that are listed at the beginning of this article."

This defect definition basically states that .NET does not correctly support the W3C's Web Services Description Language standard.

The solution to this issue is to merge the 3 API support files (2 XSD and 1 WSDL) into 1 WSDL file. This can be achieved by removing any import tags that references one of these files and then copying the contents of the referenced file into the file that had the import tag.

These 3 files are located as follows for the Training API:

- https://training-api.temando.com/schema/2009_06/server.wsdl
- https://training-api.temando.com/schema/2009_06/server.xsd
- https://training-api.temando.com/schema/2009_06/common.xsd

or these 3 files are located as follows for the Live API:

- https://api.temando.com/schema/2009_06/server.wsdl
- https://api.temando.com/schema/2009_06/server.xsd
- https://api.temando.com/schema/2009_06/common.xsd

5.4.2. Adapter Issue – Authentication

If you receive an authentication error message this may be due to the endpoint behaviours count exceeding a number of 1.

The solution to this issue is perform a comparison on the count to see that the count is greater than 1, rather than equal to 1.

Original code causing issue:

```
if (Client.Endpoint.Behaviors.Count == 1) {
```

New code with fix:

```
if (Client.Endpoint.Behaviors.Count >= 1) {
```

6. Appendix D: Insurance Guidelines

The provision of insurance by any organisation within Australia is governed by the Australian Government agency Australian Prudential Regulation Authority (<http://www.apra.gov.au/>). Strict guidelines are enforced by this agency to ensure general insurers comply with applicable requirements.

As a result of the above, insurance (by default) is not available to all API users.

To be granted access to the insurance option in the Live environment all API users must adhere to the guidelines detailed below. Once implemented a Temando representative must review those pages involved in the insurance selection process to ensure that they meet these guidelines. If all pages conform to these guidelines insurance will be enabled for the API user.

6.1. Guidelines

The insurance requirements for an API user will vary dependant on the type of account setup. The two types of accounts are Client and Affiliate.

6.1.1. Client:

With a client account they will be the Seller who is providing the goods for sale.

During the process for selecting a shipping option the option to select insurance must be displayed to the buyer in a clearly separated section.

This will be displayed as a checkbox option (automatically selected – if not then it must be approved by Temando before release) with the text “I request delivery insurance and agree to pay an extra 1.1% of the total price”. Below this checkbox the following statement must be displayed. Note: this text may be displayed within a textarea to optimise space:

For your convenience, the company arranging the transport of your newly purchased item provides you with insurance for a nominal fee of 1.1% of your purchase price (inclusive of GST).

Why should you consider buying this insurance?

Although every care is taken by the transport company whilst transporting and storing the item being transacted, it may still become lost or damaged.

In some instances, the transport company may be required to compensate you for the loss of or damage to the item. However, there are many circumstances in which they may not be required to compensate you, such as when the loss or damage is caused by an act of God, or other events beyond their control. This insurance assists to fill this gap.

Whilst every effort has been made to insure your item, you should understand that the following items are not insured (this is a summary of the exclusions and you should refer to the Terms and Conditions for a complete understanding):

- Money, deeds, securities, treasury notes, tickets, vouchers, stamps, duty stamps, any other cash equivalents,*
- Designs and/or patterns and/or plans and/or manuscripts and/or any other documents,*
- Electronic data or computer software stored on computers or hardware*
- Any living creature*
- Precious Metals, Precious Stones or Semi-precious Stones that are sent by themselves, i.e. this does not mean fixed items, e.g. rings, earrings),*
- Motor Vehicles and boats (note, this does not mean car parts, which can be covered).*

Please also note that the insurance only covers items where the origin address and the delivery address are both located in Australia. In other words, if you are purchasing your item from a country that is not Australia, then there is no insurance. Conversely, if you are resident in a country other than Australia, but buying the item from Australia the insurance will not apply either.

The price of the insurance must be clearly displayed and separate to the quote prices and total prices. This may be dynamic based on the selection for insurance (ie. checkbox). For example:

Price of Item:	\$100.00
Freight Cost:	\$10.00
Insurance Cost:	\$1.21

Total Cost: \$111.21

When the Buyer has proceeded to the payment page the following must be displayed.

The price of the insurance must be clearly displayed and separate to the quote prices and total prices. For example:

Price of Item: \$100.00

Freight Cost: \$10.00

Insurance Cost: \$1.21

Total Cost: \$111.21

A clearly separated section must also be displayed for the following insurance details.

The following text must be displayed:

Please click on all the following document names to review the [Product Disclosure Statement and wording](#), the [Financial Services Guide](#) and the [Privacy Statement](#) and then tick the box below to confirm that you have read the [Product Disclosure Statement and wording](#), the [Financial Services Guide](#) and the [Privacy Statement](#) and will take advantage of this offer.

Note: For the [blue coloured](#) text above a hyper link must be provided to the URL addresses as follows:

[Product Disclosure Statement and wording:](#)

<http://www.temando.com/downloads/Product%20Disclosure%20Statement%20and%20Wording.pdf>

[Financial Services Guide:](#)

<http://www.temando.com/downloads/Financial%20Services%20Guide.pdf>

[Privacy Statement:](#)

<http://www.temando.com/downloads/Privacy%20Statement.pdf>

Below this will be displayed a checkbox option (unselected by default) with the text “I have read and understood the terms and conditions provided above”.

Restrictions will be applied that if the checkbox is unselected then the Seller will be unable to proceed (ie. error message will be displayed stating that this must be selected).

6.1.2. Affiliate

With an affiliate account they will operate as a third party between the Seller and the Buyer. As such there will be two separate setups dependant on whether it is the Buyer who will be provided with the insurance option or the Seller.

6.1.2.1. Buyer's Choice for Insurance

This setup will be applied if insurance will be provided to the buyer as an option when purchasing an item/s and freight.

What the Buyer will see:

During the purchase process the option to select insurance must be displayed to the buyer in a clearly separated section.

This will be displayed as a checkbox option (automatically selected – if not then it must be approved by Temando before release) with the text “I request delivery insurance and agree to pay an extra 1.1% of the total price”. Below this checkbox the following statement must be displayed. Note: this text may be displayed within a textarea to optimise space:

For your convenience, the company arranging the transport of your newly purchased item provides you with insurance for a nominal fee of 1.1% of your purchase price (inclusive of GST).

Why should you consider buying this insurance?

Although every care is taken by the transport company whilst transporting and storing the item being transacted, it may still become lost or damaged.

In some instances, the transport company may be required to compensate you for the loss of or damage to the item. However, there are many circumstances in which they may not be required to compensate you, such as when the loss or damage is caused by an act of God, or other events beyond their control. This insurance assists to fill this gap.

Whilst every effort has been made to insure your item, you should understand that the following items are not insured (this is a summary of the exclusions and you should refer to the Terms and Conditions for a complete understanding):

- *Money, deeds, securities, treasury notes, tickets, vouchers, stamps, duty stamps, any other cash equivalents,*
- *Designs and/or patterns and/or plans and/or manuscripts and/or any other documents,*
- *Electronic data or computer software stored on computers or hardware*
- *Any living creature*
- *Precious Metals, Precious Stones or Semi-precious Stones that are sent by themselves, i.e. this does not mean fixed items, e.g. rings, earrings),*
- *Motor Vehicles and boats (note, this does not mean car parts, which can be covered).*

Please also note that the insurance only covers items where the origin address and the delivery address are both located in Australia. In other words, if you are purchasing your item from a country that is not Australia, then there is no insurance. Conversely, if you are resident in a country other than Australia, but buying the item from Australia the insurance will not apply either.

For terms and conditions, please review the [Product Disclosure Statement and wording](#).

Note: The “[Product Disclosure Statement and wording](#)” text in the above statement must link to the URL address <http://www.temando.com/downloads/Product%20Disclosure%20Statement%20and%20Wording.pdf>. If using a text area for the above text then the final sentence will need to be displayed external to this textarea to ensure that the hyper link is available.

The price of the insurance must be clearly displayed and separate to the quote prices and total prices. This may be dynamic based on the selection for insurance (ie. checkbox). For example:

Price of Item:	\$100.00
Freight Cost:	\$10.00
Insurance Cost:	\$1.21
Total Cost:	\$111.21

What the Seller will see:

The price of the insurance must be clearly displayed and separate to the quote prices and total prices. This may be dynamic based on the selection for insurance (ie. checkbox). For example:

Freight Cost:	\$10.00
Insurance Cost:	\$1.21
Total Cost:	\$11.21

During the process where the seller confirms the Temando quote price, if the Buyer has nominated that they want insurance a clearly separated section for displaying insurance information should be provided.

This section will contain the text: The Buyer has indicated that they would like their shipment insured.

Below this the total goods value should be displayed next to the text “Total goods value”.

Below this the following text must be displayed:

Please click on all the following document names to review the [Product Disclosure Statement and wording](#), the [Financial Services Guide](#) and the [Privacy Statement](#) and then tick the box below to confirm that you have read the [Product Disclosure Statement and wording](#), the [Financial Services Guide](#) and the [Privacy Statement](#) and will take advantage of this offer.

Note: For the [blue coloured](#) text above a hyper link must be provided to the URL addresses as follows:

[Product Disclosure Statement and wording:](#)

<http://www.temando.com/downloads/Product%20Disclosure%20Statement%20and%20Wording.pdf>

[Financial Services Guide:](#)

<http://www.temando.com/downloads/Financial%20Services%20Guide.pdf>

[Privacy Statement:](#)

<http://www.temando.com/downloads/Privacy%20Statement.pdf>

Below this will be displayed a checkbox option (unselected by default) with the text “I have read and understood the terms and conditions provided above”.

If the Seller unselects the checkbox then the following will be displayed. Another checkbox (unselected) will be displayed with the text “I understand that the buyer has requested insurance and may have paid for it already, however the buyer and I have since mutually agreed to send without insurance.”

Restrictions will be applied that if the second checkbox is unselected then the Seller will be unable to proceed (ie. error message will be displayed stating that this must be selected).

6.1.2.2. Seller's Choice for Insurance:

This setup will be applied if insurance will not be provided to the buyer as an option when purchasing an item/s and freight, and the price of insurance will be included (if nominated by the Seller) in quotes provided to the Buyer.

What the Buyer will see:

The Buyer will only see a single quote price which will either be inclusive or exclusive of insurance. Whether insurance is included will be nominated by the Seller in their settings.

What the Seller will see:

The price of the insurance (if included in Buyer's price) must be clearly displayed and separate to the quote prices and total prices. This may be dynamic based on the selection for insurance (ie. checkbox). For example:

Freight Cost:	\$10.00
Insurance Cost:	\$1.21
Total Cost:	\$11.21

If insurance was included in the Buyer's price a clearly separated section for displaying insurance information should be provided.

Next to the text "Insurance:" the following statement must be displayed. Note: this text may be displayed within a textarea to optimise space:

For your convenience, you have nominated to include insurance for the transport of your sold item for a nominal fee of 1.1% of the purchase price (inclusive of GST).

Why should you consider buying this insurance?

Although every care is taken by the transport company whilst transporting and storing the item being transacted, it may still become lost or damaged.

In some instances, the transport company may be required to compensate you for the loss of or damage to the item. However, there are many circumstances in which they may not be required to compensate you, such as when the loss or damage is caused by an act of God, or other events beyond their control. This insurance assists to fill this gap.

Whilst every effort has been made to insure your item, you should understand that the following items are not insured (this is a summary of the exclusions and you should refer to the Terms and Conditions for a complete understanding):

- Money, deeds, securities, treasury notes, tickets, vouchers, stamps, duty stamps, any other cash equivalents,*
- Designs and/or patterns and/or plans and/or manuscripts and/or any other documents,*
- Electronic data or computer software stored on computers or hardware*
- Any living creature*
- Precious Metals, Precious Stones or Semi-precious Stones that are sent by themselves, i.e. this does not mean fixed items, e.g. rings, earrings),*
- Motor Vehicles and boats (note, this does not mean car parts, which can be covered).*

Please also note that the insurance only covers items where the origin address and the delivery address are both located in Australia. In other words, if you are sending your item to a country that is not Australia, then there is no insurance. Conversely, if you are resident in a country other than Australia, but sending the item to Australia the insurance will not apply either.

Below this a checkbox will be displayed (unselected) with the following text next to it:

Please click on all the following document names to review the Product Disclosure Statement and wording, the Financial Services Guide and the Privacy Statement and then tick the box below to confirm that you have read the [Product Disclosure Statement and wording](#), the [Financial Services Guide](#) and the [Privacy Statement](#) and will take advantage of this offer.

Note: For the blue coloured text above a hyper link must be provided to the URL addresses as follows:

Product Disclosure Statement and wording:

<http://www.temando.com/downloads/Product%20Disclosure%20Statement%20and%20Wording.pdf>

Financial Services Guide:

<http://www.temando.com/downloads/Financial%20Services%20Guide.pdf>

Privacy Statement:

<http://www.temando.com/downloads/Privacy%20Statement.pdf>

Restrictions will be applied that if the checkbox is unselected then the Seller will be unable to proceed (ie. error message will be displayed stating that this must be selected).