Oracle Field Service Cloud
Integrating with Parts Catalog API

19C
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>i</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Document Purpose</td>
<td>1</td>
</tr>
<tr>
<td>Scope of the Document</td>
<td>1</td>
</tr>
<tr>
<td>Target Audience</td>
<td>1</td>
</tr>
<tr>
<td>Accessing the APIs</td>
<td>1</td>
</tr>
<tr>
<td>Glossary</td>
<td>2</td>
</tr>
<tr>
<td>2 Overview</td>
<td>3</td>
</tr>
<tr>
<td>Parts Catalog API Overview</td>
<td>3</td>
</tr>
<tr>
<td>3 Methods</td>
<td>5</td>
</tr>
<tr>
<td>Parts Catalog API Methods</td>
<td>5</td>
</tr>
<tr>
<td>4 Authentication Standards</td>
<td>7</td>
</tr>
<tr>
<td>User Authentication Structure</td>
<td>7</td>
</tr>
<tr>
<td>5 Permission</td>
<td>9</td>
</tr>
<tr>
<td>Parts Catalog API Permission</td>
<td>9</td>
</tr>
<tr>
<td>6 Catalog Creation</td>
<td>11</td>
</tr>
<tr>
<td>'create_catalog' Method</td>
<td>11</td>
</tr>
<tr>
<td>7 Catalog Methods</td>
<td>15</td>
</tr>
<tr>
<td>Catalog Upload Methods Description</td>
<td>15</td>
</tr>
<tr>
<td>8 Transactions</td>
<td>25</td>
</tr>
<tr>
<td>'get_transactions_list' Method</td>
<td>25</td>
</tr>
</tbody>
</table>
9 Catalog Management

'search_catalog' Method .................................................. 29
'upload_weights' Method ............................................... 32
'delete_weights' Method ............................................... 34
'update_types' Method .................................................. 35
'delete_catalog' Method ................................................ 38
Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit My Oracle Support or visit Accessible Oracle Support if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides. Please take one of the following surveys:

- For web-based user guide, Web-based User Guide Survey
- For tutorial feedback, Tutorial Survey
1 Introduction

Document Purpose

The document is intended to provide description of methods and SOAP transactions used in the Parts Catalog API and to ensure its successful interaction with the Clients’ external systems.

Scope of the Document

This document describes the API methods used by the Oracle Field Service Cloud Parts Catalog to exchange information (send requests and accept responses) with external systems. The overview of the actual operation of Oracle Field Service Cloud Parts Catalog and its GUI is provided in a separate dedicated document.

Target Audience

The document is intended for developers and programmers working with the Oracle Field Service Cloud Parts Catalog in order to integrate Oracle Field Service Cloud with external systems.

Accessing the APIs

To access the Oracle Field Service Cloud APIs, you must use the https://api.etadirect.com URL scheme. All old URL schemes such as, companyname.etadirect.com, na.etadirect.com, eu.etadirect.com, and so on are deprecated for Oracle Field Service Cloud versions 15.8 and later.

To access the Oracle Field Service Cloud Parts Catalog API, use the following URL:

https://api.etadirect.com/soap/parts-catalog/v2/?wsdl

Note: The Parts Catalog API v1 is removed in the Oracle Field Service Cloud 17.8 release.
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other</td>
</tr>
</tbody>
</table>
| Company               | 1) Legal entity, using Oracle Field Service Cloud  
                          2) Entity that represents a Client in Oracle Field Service Cloud system; company is created by Oracle during the process of implementation |
| Inventory             | Equipment that can be installed or deinstalled during an activity                                                                         |
| Property              | Field and field value assigned to an entity in Oracle Field Service Cloud (to user, resource, activity or inventory). There are fields and company-defined properties. |
| SOAP                  | Lightweight protocol for exchange of information in a decentralized, distributed environment                                             |
| SOAP Interface        | Interface used to receive requests and return responses via SOAP                                                                         |
| SOAP Client Application | Application running at the Client’s site and providing interaction with Oracle Field Service Cloud server via SOAP                    |
2 Overview

Parts Catalog API Overview

The Parts Catalog functionality has been designed as a means of remote access to the company’s spare parts index to be used by mobile personnel.

The idea behind it is that while technicians assigned installation jobs usually know fairly well which inventory they are going to install and load their trucks accordingly; however, those to perform repairs often will be able to determine the parts necessary for replacement only at the customer’s site after tests and diagnostics. Therefore, the Parts Catalog has been developed with a focus on maintenance and repair jobs. It enables the technician to look up the spare part required for replacement, discuss the replacement terms (price, period, etc.) with the customer, place an order for the spare part, find an alternative if the customer does not accept the initial offer and issue an invoice.

Oracle Field Service Cloud Parts Catalog is available as a module within the Manage and Mobility Applications and as an API ensuring integration with the customer’s external systems.
Chapter 3

Methods

Parts Catalog API Methods

Parts Catalog API provides a set of methods used to manage the Oracle Field Service Cloud Parts Catalog.

Catalogs can be created and searched using single operations (‘create_catalog’ and ‘search_catalog’, respectively). At the same time, data uploading to an existing catalog involves a series of operations (‘start_catalog_transaction’, ‘upload_catalog’, ‘commit_catalog_transaction’ and ‘rollback_catalog_transaction’). Optionally, the list of transactions for a certain catalog upload can be obtained by the ‘get_transactions_list’ method. Weights can be added or deleted by means of the ‘upload_weights’ and ‘delete_weights’ operations. Inventory types of the existing item types can be updated by the ‘update_types’ method. An entire catalog can be deleted using the ‘delete_catalog’ method.

The list of all methods that can be implemented with the API and their brief description is provided in the following table for reference. For the details of a specific method please refer to the corresponding section.

The following table describes the parts catalog API methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘create_catalog’</td>
<td>Creates a new catalog</td>
</tr>
<tr>
<td>‘start_catalog_transaction’</td>
<td>Starts an upload transaction for an existing catalog</td>
</tr>
<tr>
<td>‘upload_catalog’</td>
<td>Uploads data to an existing catalog</td>
</tr>
<tr>
<td>‘commit_catalog_transaction’</td>
<td>Closes an upload transaction for an existing catalog</td>
</tr>
<tr>
<td>‘rollback_catalog_transaction’</td>
<td>Aborts the current transaction and deletes all data uploaded during such transaction</td>
</tr>
<tr>
<td>‘search_catalog’</td>
<td>Searches for existing catalogs</td>
</tr>
<tr>
<td>‘upload_weights’</td>
<td>Uploads item weights to an existing catalog</td>
</tr>
<tr>
<td>‘delete_weights’</td>
<td>Deletes item weights from an existing catalog</td>
</tr>
<tr>
<td>‘update_types’</td>
<td>Updates inventory types of the item types in the catalog</td>
</tr>
<tr>
<td>‘delete_catalog’</td>
<td>Deletes all catalog data</td>
</tr>
<tr>
<td>‘get_transactions_list’</td>
<td>Gets the list of transactions for an existing catalog</td>
</tr>
<tr>
<td>‘create_catalog’</td>
<td>Creates a new catalog</td>
</tr>
<tr>
<td>‘start_catalog_transaction’</td>
<td>Starts an upload transaction for an existing catalog</td>
</tr>
</tbody>
</table>
4 **Authentication Standards**

**User Authentication Structure**

All API methods use the 'user' structure as authentication to determine the permissions of the Oracle Field Service Cloud client company user.

All customers can use the Client_ID and Client_Secret instead of login and password to populate the User Authentication Structure with credentials:

1. Register an application.
   a. In the Field Service Cloud Manage interface, click Configuration and select Applications.
   b. In the left pane, click the plus icon to open the New application window, specify the Application Name and Application ID, and click Submit.
   c. Under Authentication settings, select the Authenticate using Client ID/Client Secret check box.
   d. Click Save.
2. Select the application and under Authentication settings section, click Show Client ID / Client secret to view the Client ID and Client Secret.

The following table describes the Oracle Field Service Cloud SOAP authentication structure mandatory fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>string</td>
<td>current time in ISO 8601 format</td>
</tr>
<tr>
<td>company</td>
<td>string</td>
<td>case-insensitive identifier of the Client for which data is to be retrieved.</td>
</tr>
<tr>
<td>login</td>
<td>string</td>
<td>The client ID of the application.</td>
</tr>
<tr>
<td>auth_string</td>
<td>string</td>
<td>authentication hash; The value of this field must be computed as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>auth_string = SHA256(now + SHA256(CLIENT_SECRET + SHA256(CLIENT_ID))));</td>
</tr>
</tbody>
</table>

For example:

```
<user>
<now>CURRENT_TIME</now>
<login>CLIENT_ID</login>
<company>INSTANCE_NAME</company>
<auth_string>SHA256(CURRENT_TIME + SHA256(CLIENT_SECRET + SHA256(CLIENT_ID)))</auth_string>
</user>
```
Authentication

The 'user' structure is used for the request authentication. The relevant error is returned if the authentication fails.

<table>
<thead>
<tr>
<th>Number</th>
<th>Login</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>now</td>
<td>is different from the current time on the server and this difference exceeds the predefined time-window (30 minutes by default)</td>
</tr>
<tr>
<td>2</td>
<td>company</td>
<td>cannot be found in the Oracle Field Service Cloud</td>
</tr>
<tr>
<td>3</td>
<td>login</td>
<td>cannot be found for this company</td>
</tr>
<tr>
<td>4</td>
<td>application is not authorized to use this API</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>auth_string</td>
<td>when auth_string is not equal to: SHA256(now + SHA256(Client_Secret +SHA256(Client_ID)));</td>
</tr>
</tbody>
</table>

Otherwise authentication is successful and the request is processed further.
5 Permission

Parts Catalog API Permission

The user’s access to the Parts Catalog API is controlled by the special ‘Parts Catalog’ SOAP permission set in the ‘Permissions’ screen of the Manage Application.

Users for which the permission is set to ‘Show’ can use the Parts Catalog API methods.
6 Catalog Creation

'create_catalog' Method

The 'create_catalog' method is used to create a new Parts Catalog.

'create_catalog' Request

The ‘create_catalog’ request specifies:

- unique catalog label and language combination
- catalog structure description (sent as ‘field schemas’)

The following table describes the ‘create_catalog’ request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>Yes</td>
<td>string</td>
<td>name of the catalog to be displayed in GUI</td>
</tr>
<tr>
<td>field_schemas</td>
<td>Yes</td>
<td>array</td>
<td>array of ‘field_schema’ elements, each of which contains one of the fields to be set for the catalog</td>
</tr>
<tr>
<td>type_schemas</td>
<td>No</td>
<td>array</td>
<td>array of ‘type_schema’ elements, each of which contains an item type and may also contain the inventory type corresponding to such item type</td>
</tr>
</tbody>
</table>

'field_schema' Element of 'create_catalog' Request

The ‘field_schema’ element of the ‘create_catalog’ request contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>field identifier</td>
</tr>
<tr>
<td>name</td>
<td>Yes</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>property_label</td>
<td>No</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Required</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>searchable</td>
<td>No</td>
<td>bool</td>
<td>parameter defining whether the field is used for search</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: 0 (non-searchable)</td>
</tr>
<tr>
<td>preview</td>
<td>No</td>
<td>bool</td>
<td>parameter defining whether the field is cached in offline mode and displayed in the search results in offline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: 0 (included in search results only in the online mode)</td>
</tr>
</tbody>
</table>

`type_schema` Element of 'create_catalog' Request

The `type_schema` element of the 'create_catalog' request contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_type</td>
<td>Yes</td>
<td>string</td>
<td>item type according to the catalog</td>
</tr>
<tr>
<td>inventory_type</td>
<td>No</td>
<td>string</td>
<td>inventory type according to settings in Oracle Field Service Cloud</td>
</tr>
</tbody>
</table>

'create_catalog' Request Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <soapenv:Body>
    <ns1:create_catalog>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
      </user>
      <label>first_catalog</label>
      <language>en</language>
      <name>The name of the catalog</name>
      <field_schemas>
        <field_schema>
          <label>part_id</label>
          <name>Part number</name>
          <property_label>part_number</property_label>
          <searchable>1</searchable>
          <preview>1</preview>
        </field_schema>
        <field_schema>
          <label>vendor</label>
          <name>Vendor</name>
          <searchable>1</searchable>
          <preview>1</preview>
        </field_schema>
        <field_schema>
          <label>descr</label>
          <name>Description</name>
        </field_schema>
      </field_schemas>
    </ns1:create_catalog>
  </soapenv:Body>
</soapenv:Envelope>
```
'create_catalog' Response

The response to the 'create_catalog' request consists of the following elements describing the transaction result:

The following table describes the 'create_catalog' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
</tbody>
</table>

'create_catalog' Response Example

```xml
  xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:create_catalog_response>
      <result_code>0</result_code>
    </ns1:create_catalog_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
7 Catalog Methods

Catalog Upload Methods Description

When data is uploaded to the catalog, the whole catalog is uploaded. After the upload transaction is completed, all previous data is deleted from the catalog.

During the catalog upload the previous data can still be used (the 'search_catalog' operation is available). A started transaction can be either committed (closed with all uploaded data applied) or rolled back (aborted with all uploaded data deleted). Only one started transaction is possible for a single catalog.

'start_catalog_transaction' Method

The 'start_catalog_transaction' method is used to start a data upload transaction for a particular catalog. The method returns a unique transaction key which is to be used to commit or roll back the transaction.

'start_catalog_transaction' Request

The 'start_catalog_transaction' request must specify:

- label and language of the catalog to which data is to be uploaded

The following table describes the 'start_catalog_transaction' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog to which data is to be uploaded</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog to which data is to be uploaded</td>
</tr>
</tbody>
</table>

'start_catalog_transaction' Request Example

```xml
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <ns1:start_catalog_transaction>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
      </user>
      <label>first_catalog</label>
      <language>en</language>
    </ns1:start_catalog_transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'start_catalog' Response Example

```
    xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:start_catalog_transaction_response>
      <result_code>0</result_code>
      <transaction_key>e7ae407bc7465f4c1266bdfbfa521fa</transaction_key>
    </ns1:start_catalog_transaction_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'start_catalog_transaction' Response

The 'start_catalog_transaction' response returns a unique transaction key which is to be used to commit or roll back the transaction.

The following table describes the 'start_catalog_transaction' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>transaction_key</td>
<td>string</td>
<td>unique key of the transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is 0.</td>
</tr>
</tbody>
</table>

'upload_catalog' Method

The 'upload_catalog' method uploads data to the selected catalog. During the upload transaction (before its commitment) only the data uploaded in the previous committed transaction is available for search. The uploaded items become available only after the transaction is committed.

'upload_catalog' Request

The 'upload_catalog' request must specify:

- transaction key obtained as the result of the 'start_catalog_transaction' operation
- items to be uploaded to the catalog and their properties

The following table describes the 'upload_catalog' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>Name</td>
<td>Required</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>transaction_key</td>
<td>Yes</td>
<td>string</td>
<td>array of 'item' elements each of which contains properties of a single item to be uploaded to the catalog</td>
</tr>
<tr>
<td>catalog</td>
<td>Yes</td>
<td>array</td>
<td>array of 'item' elements each of which contains properties of a single item to be uploaded to the catalog</td>
</tr>
</tbody>
</table>

'item' Element of 'upload_catalog' Request
The 'item' element of the 'upload_catalog' request consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Yes</td>
<td>string</td>
<td>item type</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>item identifier</td>
</tr>
<tr>
<td>weight</td>
<td>No</td>
<td>int</td>
<td>item weight used for items sorting during cache loading. Items with greater weight are the first to be loaded in the cache. default value: 0</td>
</tr>
<tr>
<td>fields</td>
<td>Yes</td>
<td>array</td>
<td>array of 'field' elements, each of which contains one of the item fields and its value</td>
</tr>
<tr>
<td>linked_items</td>
<td>No</td>
<td>array</td>
<td>array of 'linked_item' elements, each containing an alternative item with which the item defined in the 'item' element can be substituted and the information on such alternative item</td>
</tr>
<tr>
<td>images</td>
<td>No</td>
<td>array</td>
<td>array of 'image' elements, each containing a URL of the item image</td>
</tr>
</tbody>
</table>

- 'field' Element of 'upload_catalog' Request
The 'field' element of the 'upload_catalog' request consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>field identifier</td>
</tr>
<tr>
<td>value</td>
<td>Yes</td>
<td>string</td>
<td>field value</td>
</tr>
</tbody>
</table>

- 'linked_item' Element of 'upload_catalog' Request
The 'linked_item' element of the 'upload_catalog' request consists of the following fields:
### Catalog Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>unique linked item identifier</td>
</tr>
</tbody>
</table>

| display_data | No       | string | text comments to the linked item to be displayed in GUI |

- **'image' Element of 'upload_catalog' Request**

The ‘image’ element of the ‘upload_catalog’ request consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_url</td>
<td>Yes</td>
<td>string</td>
<td>valid URL address of the image associated with the item defined in the 'item' element</td>
</tr>
</tbody>
</table>

**Note:** if the request contains several items with the same label, data of the last such item is used. If the request contains several fields with the same label, data of the last such field is used. If the request contains several linked items with the same label, data of the last such linked item is used.

**'upload_catalog' Request Example**

```xml
http://schemas.xmlsoap.org/soap/envelope/ xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:upload_catalog>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
      </user>
      <transaction_key>e7ae407bc7465f4c1266bdfbea521fa</transaction_key>
      <catalog>
        <item>
          <type>parts</type>
          <label>RG5-7691-250CN</label>
          <weight>12</weight>
          <fields>
            <field>
              <label>vendor</label>
              <value>HEWLETT PACKARD</value>
            </field>
            <field>
              <label>descr</label>
              <value>110V FUSER ASSEMBLY</value>
            </field>
            <field>
              <label>price</label>
              <value>943.80</value>
            </field>
          </fields>
          <linked_items>
            <linked_item>
              <label>RG5-7691-250CN</label>
              <display_data>1</display_data>
            </linked_item>
            <linked_item>
              <label>RG5-7691-250CF</label>
              <display_data>2</display_data>
            </linked_item>
          </linked_items>
        </item>
        <item>
          <type>parts</type>
          <label>RG5-7691-250CN</label>
          <weight>12</weight>
          <fields>
            <field>
              <label>vendor</label>
              <value>HEWLETT PACKARD</value>
            </field>
            <field>
              <label>descr</label>
              <value>110V FUSER ASSEMBLY</value>
            </field>
            <field>
              <label>price</label>
              <value>943.80</value>
            </field>
          </fields>
          <linked_items>
            <linked_item>
              <label>RG5-7691-250CN</label>
              <display_data>1</display_data>
            </linked_item>
            <linked_item>
              <label>RG5-7691-250CF</label>
              <display_data>2</display_data>
            </linked_item>
          </linked_items>
        </item>
      </catalog>
    </ns1:upload_catalog>
  </SOAP-ENV:Body>
</http://schemas.xmlsoap.org/soap/envelope/>
```
`upload_catalog` Response

The `upload_catalog` response returns the count of items successfully uploaded in the current operation, the count of items not uploaded due to invalid data structure and the labels of not uploaded items.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. <code>result_code</code> is always returned in a response. '0' on success. If the transaction fails, the <code>result_code</code> is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>uploaded_count</td>
<td>int</td>
<td>number of successfully uploaded items. Returned only when</td>
</tr>
<tr>
<td>not_uploaded_count</td>
<td>int</td>
<td>number of not uploaded items. Returned only when</td>
</tr>
<tr>
<td>not_uploaded_labels</td>
<td>array</td>
<td>array of 'label' fields containing labels of item not uploaded in the transaction. Returned only when</td>
</tr>
</tbody>
</table>

'field_schema' Element of 'not_updated_labels' Array of 'update_catalog' Response

The 'not_updated_labels' array of the 'update_catalog' response contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>string</td>
<td>labels of items not uploaded in the transaction due to invalid data structure</td>
</tr>
</tbody>
</table>

'upload_catalog' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:upload_catalog_response>
      <result_code>0</result_code>
      <uploaded_count>1</uploaded_count>
      <not_uploaded_count>1</not_uploaded_count>
      <not_uploaded_labels>
        <label>H-AZ240135</label>
      </not_uploaded_labels>
    </ns1:upload_catalog_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'commit_catalog_transaction' Method

The 'commit_catalog_transaction' method is used to close an active transaction and apply all uploaded data. After the transaction is committed, the uploaded items become available for search.

'commit_catalog_transaction' Request

The 'commit' request must specify:

- transaction key obtained as the result of the 'start_catalog_transaction' operation

The following table describes the 'commit_catalog_transaction' request parameters.
### Catalog Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>transaction_key</td>
<td>Yes</td>
<td>string</td>
<td>unique transaction key obtained as the result of the 'start_catalog_transaction' operation</td>
</tr>
</tbody>
</table>

#### 'commit_catalog_transaction' Request Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:commit_catalog_transaction>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
      </user>
      <transaction_key>e7ae407bc7465f4c1266bdfbfaa521fa</transaction_key>
    </ns1:commit_catalog_transaction>
  </SOAP-ENV:Body>
</soapenv:Envelope>
```

#### 'commit_catalog_transaction' Response

The response to the 'commit_catalog_transaction' request consists of the following elements describing the transaction result:

The following table describes the 'commit_catalog_transaction' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
</tbody>
</table>

#### 'commit_catalog_transaction' Response Example

```xml
http://schemas.xmlsoap.org/soap/envelope/xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:commit_catalog_transaction_response>
      <result_code>0</result_code>
    </ns1:commit_catalog_transaction_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'rollback_catalog_transaction' Method

The 'rollback_catalog_transaction' operation is used to abort an active transaction and delete all data uploaded in such transaction. The catalog continues using the previous data.

'rollback_catalog_transaction' Request

The 'rollback' request must specify:

- transaction key obtained as the result of the 'start_catalog_transaction' operation

The following table describes the 'rollback_catalog_transaction' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>transaction_key</td>
<td>Yes</td>
<td>string</td>
<td>unique transaction key obtained as the result of the 'start_catalog_transaction' operation</td>
</tr>
</tbody>
</table>

'rollback_catalog_transaction' Request Example

```xml
http://schemas.xmlsoap.org/soap/envelope/xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:rollback_catalog_transaction>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc0712b404a8d870e69</auth_string>
      </user>
      <transaction_key>e7ae407bc7465f4c1266bdfbfee521fa</transaction_key>
    </ns1:rollback_catalog_transaction>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'rollback_catalog_transaction' Response

The response to the 'rollback_catalog_transaction' request consists of the following elements describing the transaction result:

The following table describes the 'rollback_catalog_transaction' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
</tbody>
</table>
### Catalog Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

#### 'rollback_catalog_transaction' Response Example

```xml
xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:rollback_catalog_transaction_response>
      <result_code>0</result_code>
    </ns1:rollback_catalog_transaction_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
8 Transactions

'get_transactions_list' Method

The 'get_transactions_list' method is used to get the list of upload transactions for the selected catalog. This method provides information about the catalog upload history. At the same time, it allows to view the list of active transactions along with their transaction keys.

'get_transactions_list' Request

The 'get_transactions_list' request must specify:

- label and language of the catalog the transaction list is to be obtained for

The following table describes the 'get_transactions_list' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog the transaction list is requested for</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog the transaction list is requested for</td>
</tr>
<tr>
<td>status</td>
<td>No</td>
<td>string</td>
<td>status of the catalog transactions in a response (i.e. 'in_progress', 'committed', 'rolled_back')</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>if this parameter is not set, the transactions with all statuses will be included in a response</td>
</tr>
<tr>
<td>from</td>
<td>No</td>
<td>int</td>
<td>the position of the transaction in the list, starting from which the transactions are to be selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the default value of this parameter is '0'</td>
</tr>
<tr>
<td>count</td>
<td>No</td>
<td>int</td>
<td>count of transactions in a response</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the default value of this parameter is '20'</td>
</tr>
</tbody>
</table>

'get_transactions_list' Request Example

```
http://schemas.xmlsoap.org/soap/envelope/xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:get_transactions_list>
      <user>
        <now>2010-11-19T13:50:10Z</now>
        <company>sunrise</company>
        <login>soap</login>
        <auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
    </user>
  </ns1:get_transactions_list>
</SOAP-ENV:Body>
```
'get_transactions_list' Response

The response to the 'get_transactions_list' request consists of the following elements describing the list of found catalog transactions:

The following table describes the 'get_transactions_list' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response. '0' on success. If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>total</td>
<td>int</td>
<td>the total count of transactions</td>
</tr>
<tr>
<td>from</td>
<td>int</td>
<td>the position of the transaction in the list, starting from which the transactions are to be selected the default value of this parameter is 0</td>
</tr>
<tr>
<td>count</td>
<td>int</td>
<td>The actual number of transactions returned in a response</td>
</tr>
<tr>
<td>transactions</td>
<td>array</td>
<td>list of transactions found</td>
</tr>
</tbody>
</table>

'transaction' Element of 'get_transactions_list' Response

The 'transaction' element of the 'get_transactions_list' response contains the following fields describing a transaction within the transactions list:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>catalog_label</td>
<td>string</td>
<td>label of the catalog for which the current transaction has been performed</td>
</tr>
<tr>
<td>catalog_language</td>
<td>string</td>
<td>language of the catalog for which the current transaction has been performed</td>
</tr>
<tr>
<td>transaction_key</td>
<td>string</td>
<td>unique key of the transaction obtained as the result of the 'start_catalog_transaction' operation</td>
</tr>
<tr>
<td>items_count</td>
<td>int</td>
<td>number of items uploaded within the current transaction</td>
</tr>
<tr>
<td>started</td>
<td>DateTime</td>
<td>the start date/time of the current transaction</td>
</tr>
</tbody>
</table>
## Transactions

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>completed</td>
<td>DateTime</td>
<td>the end date/time of the current transaction</td>
</tr>
<tr>
<td>status</td>
<td>string</td>
<td>status of the catalog transactions (i.e. 'in_progress', 'committed', 'rolled_back')</td>
</tr>
</tbody>
</table>

### 'get_transactions_list' Response Example

```xml
xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:get_transactions_list_response>
      <result_code>0</result_code>
      <total>2</total>
      <count>2</count>
      <from>0</from>
      <transactions>
        <transaction>
          <catalog_label>first_catalog</catalog_label>
          <catalog_language>en</catalog_language>
          <transaction_key>cae1e6efdd38875c91eea05b260b8383</transaction_key>
          <items_count>3</items_count>
          <started>2014-03-05 14:51:43</started>
          <completed>2014-03-05 14:53:43</completed>
          <status>committed</status>
        </transaction>
        <transaction>
          <catalog_label>first_catalog</catalog_label>
          <catalog_language>en</catalog_language>
          <transaction_key>ff4fb3cf681d643aa49219d348244c2b</transaction_key>
          <items_count>0</items_count>
          <started>2014-03-07 14:51:43</started>
          <completed></completed>
          <status>in_progress</status>
        </transaction>
      </transactions>
    </ns1:get_transactions_list_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
9 Catalog Management

'search_catalog' Method

The 'search_catalog' method is used to search for the existing Parts Catalogs.

'search_catalog' Request

The 'search_catalog' request may specify:

- catalog label and language, if needed

The following table describes the 'search_catalog' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog to search for</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog to search for</td>
</tr>
</tbody>
</table>

'search_catalog' Request Example

```
<SOAP-ENV:Body>
<ns1:search_catalog>
<user>
<now>2010-11-19T13:50:10Z</now>
<company>sunrise</company>
<login>soap</login>
<auth_string>7160f16f31540cc07128404a8d878e69</auth_string>
</user>
<label>first_catalog</label>
<language>en</language>
</ns1:search_catalog>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'search_catalog' Response

The 'search_catalog' response contains the list of catalogs with their description and structure as well as the total number of items within. If a transaction is in progress for a catalog, the items uploaded during such transaction are not taken into account calculation of the total. During items upload to a newly created catalog, the operation returns an empty catalog ("items_count" = 0). During items upload to a catalog which already contained items before, only the previous items count is returned.
**Note:** only the items having 'weight' > 0 are counted.

The following table describes the 'search_catalog' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>catalogs</td>
<td>array</td>
<td>array of 'catalog' elements, each containing data of a single catalog</td>
</tr>
</tbody>
</table>

'catalog' Element of 'upload_catalog' Request

The 'catalog' element of the 'search_catalog' response consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>string</td>
<td>technical name of the catalog</td>
</tr>
<tr>
<td>language</td>
<td>string</td>
<td>language of the catalog</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>name of the catalog to be displayed in GUI</td>
</tr>
<tr>
<td>items_count</td>
<td>int</td>
<td>number of items in the catalog with 'weight' greater than 0</td>
</tr>
<tr>
<td>field_schemas</td>
<td>array</td>
<td>array of 'field_schema' elements, each containing one of the fields to be set for the catalog</td>
</tr>
<tr>
<td>type_schemas</td>
<td>array</td>
<td>array of 'type_schema' elements, each containing an item type and, optionally may also contain the inventory type corresponding to such item type</td>
</tr>
</tbody>
</table>

- 'field_schema' Element of 'search_catalog' Request

The 'field_schema' element of the 'search_catalog' response consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>string</td>
<td>field identifier</td>
</tr>
</tbody>
</table>
### Chapter 9: Catalog Management

#### Name | Type | Description
--- | --- | ---
name | string | field name to be displayed in GUI
property_label | string | identifier of the inventory property to transfer data from the catalog to the inventory form in Oracle Field Service Cloud
searchable | bool | parameter defining whether the field is used for search
default value: 0 (non-searchable)
preview | bool | parameter defining whether the field is cached in offline mode and displayed in the search results in offline
default value: 0 (included in search results only in the online mode)

- **'type_schema' Element of 'search_catalog' Request**

The 'type_schema' element of the 'search_catalog' response consists of the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_type</td>
<td>string</td>
<td>item type according to the catalog</td>
</tr>
<tr>
<td>inventory_type</td>
<td>string</td>
<td>inventory type according to settings in Oracle Field Service Cloud</td>
</tr>
</tbody>
</table>

**'search_catalog' Response Example**

```xml
  <SOAP-ENV:Body>
    <ns1:search_catalog_response>
      <result_code>0</result_code>
      <catalogs>
        <catalog>
          <label>first_catalog</label>
          <language>en</language>
          <name>The name of the catalog</name>
          <items_count>485019</items_count>
          <field_schemas>
            <field_schema>
              <label>vendor</label>
              <name>Vendor</name>
              <searchable>1</searchable>
              <preview>1</preview>
            </field_schema>
            <field_schema>
              <label>descr</label>
              <name>Description</name>
              <searchable>1</searchable>
            </field_schema>
            <field_schema>
              <label>price</label>
              <name>Cost</name>
            </field_schema>
          </field_schemas>
        </catalog>
      </catalogs>
    </ns1:search_catalog_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'upload_weights' Method

The 'upload_weights' method is used to upload weights for the catalog items. Weight is an optional parameter used for items sorting during cache loading. Items with greater weight are the first to be loaded to the cache. If the weight label of a particular item already exists, the 'upload_weights' operation overwrites the previous weight value with the new one.

The 'upload_weights' operation must be performed after the items have been already uploaded to the catalog. If the catalog items are updated, their weights are removed in the process, because the weights are associated with the old items.

'upload_weights' Request

The 'upload_weights' request must specify:

- label and language of the catalog to which weights are to be uploaded
- labels of the items for which weights are to be uploaded
- weight labels and values

The following table describes the 'upload_weights' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog to which weights are to be uploaded</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog to which weights are to be uploaded</td>
</tr>
<tr>
<td>name</td>
<td>Yes</td>
<td>string</td>
<td>name of the catalog to be displayed in GUI</td>
</tr>
<tr>
<td>weights</td>
<td>Yes</td>
<td>array</td>
<td>array of 'weight' elements, each of which contains weight to be uploaded for the specified item</td>
</tr>
</tbody>
</table>
'weight' Element of 'upload_weights' Request

The 'weight' element of the 'upload_weights' request contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_label</td>
<td>Yes</td>
<td>string</td>
<td>unique identifier of the item to which weight is to be uploaded</td>
</tr>
<tr>
<td>weight_label</td>
<td>Yes</td>
<td>string</td>
<td>unique identifier of the weight to be uploaded for the specified item</td>
</tr>
<tr>
<td>weight</td>
<td>Yes</td>
<td>int</td>
<td>value of the weight to be uploaded for the specified item</td>
</tr>
</tbody>
</table>

'upload_weights' Request Example

```
http://schemas.xmlsoap.org/soap/envelope/xmlns:ns1="urn:toatech:PartsCatalog:1.0">
<SOAP-ENV:Body>
<ns1:upload_weights>
<user>
<now>2013-12-25T11:37:40Z</now>
<login>soap</login>
<company>in429</company>
<auth_string>3b3ec291807984a39e0e2a1f4d5aae21</auth_string>
</user>
<label>tttest2</label>
<language>en</language>
<weights>
<weight>
<item_label>test_item1</item_label>
<weight>7</weight>
<weight_label>weight_label1</weight_label>
</weight>
<weight>
<item_label>test_item1</item_label>
<weight>12</weight>
<weight_label>weight_label2</weight_label>
</weight>
<weight>
<item_label>test_item2</item_label>
<weight>3</weight>
<weight_label>weight_label1</weight_label>
</weight>
<weight>
<item_label>test_item3</item_label>
<weight>4</weight>
<weight_label>weight_label1</weight_label>
</weight>
</weights>
</ns1:upload_weights>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'upload_weights' Response

The 'upload_weights' response returns the count of weights successfully uploaded in the operation.
The following table describes the ‘upload_weights’ response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. ‘result_code’ is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>’0’ on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the ‘result_code’ is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the ‘result_code’ is other than 0.</td>
</tr>
<tr>
<td>uploaded_count</td>
<td>int</td>
<td>number of successfully uploaded weights.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when</td>
</tr>
</tbody>
</table>

‘upload_weights’ Response Example

```xml
  xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:upload_weights_response>
      <result_code>0</result_code>
      <uploaded_count>4</uploaded_count>
    </ns1:upload_weights_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

‘delete_weights’ Method

The ‘delete_weights’ method is used to delete all weights set for the items of the specified catalog.

‘delete_weights’ Request

The ‘delete_weights’ request must specify:

- label and language of the catalog from which weights are to be deleted

The following table describes the ‘delete_weights’ request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog from which weights are to be deleted</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog from which weights are to be deleted</td>
</tr>
</tbody>
</table>
'delete_weights' Request Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <soapenv:Body>
    <ns1:delete_weights>
      <user>
        <now>2013-12-25T13:41:21Z</now>
        <login>soap</login>
        <company>in429</company>
        <auth_string>cf368aa2307a55112f640b449bf7fe46</auth_string>
      </user>
      <label>tttest2</label>
      <language>en</language>
    </ns1:delete_weights>
  </soapenv:Body>
</SOAP-ENV:Envelope>
```

'delete_weights' Response

The response to the 'delete_weights' request consists of the following elements describing the transaction result:

The following table describes the 'delete_weights' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
</tbody>
</table>

'delete_weights' Response Example

```xml
    xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:delete_weights_response>
      <result_code>0</result_code>
    </ns1:delete_weights_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'update_types' Method

The 'update_types' method is used to update item types belonging to a particular Parts Catalog.
As the result of the 'update_types' operation, the inventory type labels of the specified item types are updated. If an item type sent in the request cannot be found in the Parts Catalog database, such item is skipped. If the inventory type label sent in the request for a particular item type is the same as the inventory type label existing for the same item type, such item type is not updated.

'update_types' Request

The 'update_types' request specifies:

- unique catalog label and language combination
- item types for which inventory type labels are to be updated
- new values of the inventory type labels

The following table describes the 'update_types' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog to be created</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog to be created</td>
</tr>
<tr>
<td>type_schemas</td>
<td>No</td>
<td>array</td>
<td>array of 'type_schema' elements, each of which contains an item type and may also contain the inventory type corresponding to such item type</td>
</tr>
</tbody>
</table>

'type_schema' Element of 'update_types' Request

The 'type_schema' element of the 'update_types' request contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_type</td>
<td>Yes</td>
<td>string</td>
<td>item type label according to the catalog</td>
</tr>
<tr>
<td>inventory_type</td>
<td>No</td>
<td>string</td>
<td>inventory type label according to settings in Oracle Field Service Cloud</td>
</tr>
</tbody>
</table>

'update_types' Request Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <soapenv:Body>
    <ns1:update_types>
      <user>
        <login>soap</login>
        <company>in429</company>
        <auth_string>cf368aa2307a55112f640b449bf7fe46</auth_string>
      </user>
      <label>tttest2</label>
    </ns1:update_types>
  </soapenv:Body>
</soapenv:Envelope>
```

36
'update_types' Response

The 'update_types' response returns the count of successfully updated items and not updated items. If any items are not updated in the transaction, the response also contains the list of such type labels:

The following table describes the 'update_types' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td>updated_count</td>
<td>int</td>
<td>number of successfully updated items</td>
</tr>
<tr>
<td>not_updated_count</td>
<td>int</td>
<td>number of items not updated in the transaction due to invalid data structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or non-existent item type label</td>
</tr>
<tr>
<td>not_updated_labels</td>
<td>array</td>
<td>array of 'label' fields containing labels of item types not updated in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transaction</td>
</tr>
</tbody>
</table>

'not_updated_labels' Array of 'update_types' Response

The 'not_updated_labels' array of the 'update_types' response contains the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>string</td>
<td>labels of item types not updated in the transaction due to invalid data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure or non-existent item type label</td>
</tr>
</tbody>
</table>

'update_types' Response Example

```xml
xmlns:ns1="urn:toatech:PartsCatalog:1.0">
<SOAP-ENV:Body>
<ns1:update_types_response>
<ns1:update_type><item_type>some item type label</item_type><inventory_type>some inventory type label</inventory_type></ns1:inventory_type></ns1:update_type>
</ns1:update_types_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'delete_catalog' Method

The 'delete_catalog' method is used to delete all data of the specified catalog.

'delete_catalog' Request

The 'delete_catalog' request must specify:

- label and language of the catalog to be deleted

The following table describes the 'delete_catalog' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' node</td>
</tr>
<tr>
<td>label</td>
<td>Yes</td>
<td>string</td>
<td>technical name of the catalog from which weights are to be deleted</td>
</tr>
<tr>
<td>language</td>
<td>Yes</td>
<td>string</td>
<td>language of the catalog from which weights are to be deleted</td>
</tr>
</tbody>
</table>

'delete_catalog' Request Example

```xml
http://schemas.xmlsoap.org/soap/envelope/xmlns:ns1="urn:toatech:PartsCatalog:1.0">
  <SOAP-ENV:Body>
    <ns1:delete_weights>
      <user>
        <now>2013-12-25T13:41:21Z</now>
        <login>soap</login>
        <company>in429</company>
        <auth_string>cf368aa2307a55112f640b449bf7fe46</auth_string>
      </user>
      <label>tttest2</label>
      <language>en</language>
    </ns1:delete_weights>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'delete_catalog' Response

The response to the 'delete_catalog' request consists of the following elements describing the transaction result:

The following table describes the 'delete_catalog' response parameters.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code. 'result_code' is always returned in a response.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'0' on success.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the transaction fails, the 'result_code' is other than 0.</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>more specific description of the error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returned only when the 'result_code' is other than 0.</td>
</tr>
</tbody>
</table>

'delete_catalog' Response Example

```xml
 xmlns:ns1="urn:toatech:PartsCatalog:1.0">
 <SOAP-ENV:Body>
  <ns1:delete_catalog_response>
   <result_code>0</result_code>
  </ns1:delete_catalog_response>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```