Alice REST API Technical Documentation

September 2021/v 2.15
Revision 1

1. Introduction
The how and why of the API.

The Alice REST API, in short Alice REST, is used jointly with the Alice POS software in order to retrieve information from it for use by other software modules or 3rd party software.

Most requests made to the API have read-only privileges but some are also able to update and write data to the database or log files.

It is beyond the scope of this document to explain how a REST API functions or how Alice POS will react to transactions made using Alice REST. The purpose of this document is solely to be used as a reference about which transactions (and their arguments) are available and implemented with the API.

This document uses the standard glyphicons to represent portions of the text in it. Here's the complete legend.

Documentation related glyphicons

A note in need of attention.

A warning you should read before you proceed.

A link to an external document.

Transaction related glyphicons.

A transaction that can write to the database.

An optional transaction parameter.

A required transaction parameter.

Possible values for a parameter.

The return value of a transaction.

2. Transaction format
General formatting and guidelines with examples.

All transactions are sent and received using the XML 1.0 format. Numerous examples and tutorials are available on the internet but we recommend getting familiar with the official version of the format.

🔗W3C XML 1.0 Reference

A sample request is built using the following structure

```xml
<request>
 <credentials>
   <apikey>API_KEY</apikey>
   <timestamp>TIME_OF_REQUEST</timestamp>
   <signatureversion>VERSION_OF_SIGNATURE</signatureversion>
   <signaturemethod>ENCRYPTION_METHOD</signaturemethod>
   <signature>REQUEST_SIGNATURE</signature>
 </credentials>
 <type>TRANSACTION_TYPE</type>
 <filters>
       <filter>
             <field>FIELD_NAME</field>
             <like>LIKE_FORMAT</like>
       </filter>
 </filters>
</request>
```

A sample response is also built in the same fashion

```xml
<response>
 <request>
   <credentials>
     <apikey>API_KEY</apikey>
     <timestamp>TIME_OF_REQUEST</timestamp>
     <signatureversion>VERSION_OF_SIGNATURE</signatureversion>
     <signaturemethod>ENCRYPTION_METHOD</signaturemethod>
     <signature>REQUEST_SIGNATURE</signature>
   </credentials>
   <type>TRANSACTION_TYPE</type>
   <filters>
     <filter>
       <field>FIELD_NAME</field>
       <like>LIKE_FORMAT</like>
     </filter>
   </filters>
 </request>
 RESPONSE_BLOCK
</response>
```

✏️Each transaction response has its own structure for the RESPONSE_BLOCK. It is the return value of the computed request. The RESPONSE_BLOCK can also be an error block which contains a single <error> tag with a message in its text node.

Contents of the various RESPONSE_BLOCK are described in detail for each transaction on subsequent pages of this documentation.

2.1 Authentication
Secure transactions with the REST API

All communications with the REST API are secured with an API key and an encrypted signature key. The signature is unique to every request/response and is built with several unique values and a secret key that are encrypted with HmacSHA256. That signature is used to ensure the validity of the request/response.

<u>Parameters</u>

☑	apikey (string) Unique key to every system
☑	timestamp (timestamp) Current time and date of the request
☑	signatureversion (integer) Which version of the signature we want to use
☑	signaturemethod (string) Which encryption method to use
☑	signature (string) The unique signature hash

*** Please note that the secret key is never sent in the request/response.

3.1 Date

Get the server's date and time.

Getting the database server's date can be very useful in order to synchronize or compare the client's date and time.

Parameters

☑	type: Date (string)

Response

➦	date (string) The database server's date

Example:

```xml
<response>
 <request>
      <type>Date</type>
 </request>
 <date format="Y-m-d H:i:s">2015-02-20 17:10:06</date>
</response>
```

3.2 Categories
Get Alice's categories.

Alice products are organized in a hierarchy of categories. Getting the class ID of each category can be useful in order to place products in the corresponding category tree node.

Parameters

☑	type: Category (string)
☐	lang (integer)(deprecated – translations are always returned) Will return categories in another language than the default.
☐	sku (string) (id of the category) Will return the information of a specific category.
☐	when (timestamp) - deprecated Will respond only with categories modified after the timestamp
☐	whenmod (timestamp) Will respond only with categories modified after the timestamp.

Response

➤	id (string) ID of the category
➤	whenmod (timestamp) Time the category was last modified.
➤	ownerid (string) If the category is a child of another the parent's id will be returned
➤	name (string) Name of the category.
➤	namefr (string) Name of the category. French if there is one
➤	nameen (string) Name of the category. English if there is one
➤	namesp (string) Name of the category. Spanish if there is one
➤	desc (string) Description of the category.
➤	descfr (string) Description of the category. French if there is one
➤	descen (string) Description of the category. English if there is one
➤	descsp (string) Description of the category. Spanish if there is one
➤	level (integer) The depth of the category in the hierarchy.
➤	online (integer) If the category is to be displayed in an online store.

☰	0 = No
☰	1 = Yes

		enable (string) - deprecated
		If the category is enabled or not.
	☰	t = Yes
	☰	f = No

	order (integer)
	The sort order of the category in the current parent, smaller is first.

	Enable(bool)
	☰	t = Category is active
	☰	f = Category is not active

	picture (string)
	If the category has a picture

	picturewhen (timestamp)
	The last time a category picture was updated

Example:

```xml
<response>        <request>
            <type>Category</type>
        </request>
        <categories>
            <category>
                <id>class_442m</id>
                <whenmod>1399052931.3327</whenmod>
                <ownerid/>
                <name>MISCELLANEOUS</name>
                <desc/>
                <level>1</level>
                <online>1</online>
                <order>15</order>
                <enable>t</enable>
                <picture>atta_70464m</picture>
                <picturewhenmod>1496761807.63567</picturewhenmod>
            </category>
            ...
        </categories>
</response>
```

3.3 Products
Get a filtered list of Alice products.

Getting products if the core of the REST API. It allows 3rd party software to display and sell Alice products in their own custom software.

For a more detailed explanation of the Matrix specific fields, see section 5.2.

Parameters

☑	type: Product (string)
☑	currency (string) (LIKE) The currency used for the pricing fields. Products can have different prices in different countries.

▤	cad = Canadian Dollars
▤	usd = United States Dollars
▤	usdz1 = United States (zone 1) Dollars
▤	cop = Colombian Pesos
▤	egp = Egyptian Pounds

☐	sku (string) (LIKE) Find the product with a specific SKU (pric_##)
☐	parentsku (string) (LIKE) Find all if the variants of the specified SKU (pric_##)
☐	matrixmode (integer) (PARAM) Filter the return depending on the matrix structure

▤	0 = All – show parents and variants
▤	1 = Parents and non-variants only
▤	2 = Variants only

|	
☐	whenmod (timestamp) (AFTER) Only products modified after the timestamp will be returned. ("when" and "whenonline" can also be used with the same result)
☐	name (string) (LIKE) Only products matching the name pattern will be returned.
☐	classid (string) (LIKE) Only products that belong to the class ID pattern will be returned. Useful for getting products from a category.
☐	withimage (any value) (LIKE) Only return product with an image
☐	upccode (string) (PARAM) Finds the product with the UPC or custom code, exact match only. A list of UPCs can also be used (separated by commas) ie (upc1,upc2,upc3)
☐	withinventory (any value) (==deprecated if productfilter is used==) Only return product with in stock
☐	checkonline (any value) (==deprecated if productfilter is used==) Removes the online validation check and returns all products
☐	productfilter (integer) (PARAM) Used for a more refined and clear filtering of products to be returned based on online availability state, current inventory and past activity.

▤	1 = All – all products regardless of status and inventory

	2 = With Activity – sold or ordered even if no current inventory
	3 = With Activity Last Year –sold or ordered even if no current inventory
	4 = In-Stock – products in inventory and visible online, "In Store Only" are IN-CLUDED (online modes 1 and 3)
	5 = Online Only – products in inventory and visible online, "In Store Only" are EXCLUDED (online mode 1)
	6 = In Store Only – products in inventory that are available "In Store Only". (online mode 3)

Response

	whenmod (timestamp) The time the product was last modified.
	sku (string) The product's unique SKU
	alicecode (string) The product's Alice Code (ABC#123456)
	name (string) Name of the product to be displayed to customers.
	namefr (string) Name of the product to be displayed to customers. French if it exists
	nameen (string) Name of the product to be displayed to customers. English if it exists
	namesp (string) Name of the product to be displayed to customers. Spanish if it exists
	desc (text) Product description
	descfr (text) Product description French if it exists
	descen (text) Product description English if it exists
	descsp (text) Product description Spanish if it exists
	release (timestamp) Date and time the product was or is to be released.
	ismatrixparent (boolean) Indicated that this product has variations
	parentsku (text) SKU of the product's matrix parent
	enable (boolean) If the product is enabled or not. In Alice, products are never deleted from the database, they are disabled and can be reactivated.

	t = Yes
	f = No

	taxable (boolean) Indicates if the product is taxable or not

	t = Yes
	f = No

	online (integer) If the product is to be displayed in an online store

		0 = Not visible online
		1 = Visible online
		3 = Visible online but only available in-store

➦	sendtoecwid (boolean) Indicates that the product should be sent to the Ecwid Store (if applicable)
➦	sendtopac (boolean) Indicates that the product should be sent to the PACs Store (if applicable)
➦	prodseo (string) SEO as defined in Alice, should be use in the online store.
➦	buynew (float) Price of the product when purchasing new. (cost)
➦	buyused (float) Price of the product when purchasing used (cost)
➦	sellnew (float) Price of the product when sold new
➦	sellused (float) Price of the product when sold used
➦	size (string) Value of the shipping size attribute (dimension for shipping calculation)
➦	weight (string) Value of the shipping weight attribute (for shipping calculation)
➦	platform (string) Platform of the product or parent category
➦	category (string) The product's category name
➦	UPC (string) (deprecated - use the "upccodes" node) List of UPC(s) associated with the product separated by commas
➦	picture (string) If the product has a picture
➦	picturewhenmod (timestamp) The last time a product picture was updated
➦	variants (Node) Details of the products variations

	➦	variantids (SubNode)
		➦ variantidlist (text array) Ids (SKU) of all the product's variations
	➦	matrixconfig (SubNode) List of all attributes and available values for the variants

		➦	configoption (SubNode)
			➦ variant_namefr (text) Name of the attribute in French
			➦ variant_nameen (text) Name of the attribute in English
			➦ variant_namesp (text) Name of the attribute in Spanish
			➦ variant_id (text) Id of the attribute
			➦ variant_valuenamefr (text array) Name of the attribute values in French
			➦ variant_valuenameen (text array) Name of the attribute values in English

				➦	variant_valuenamesp (text array) Name of the attribute values in Spanish	
				➦	variant_valueids (text array) Ids of all the possible values for that attribue	

➦	promoprice (Node)		
	➦	promosnew (SubNode) Best promo price for new items	
		➦	promolinkid (string) Id of the promolink
		➦	promoid (string) Id of the parent promotion (or sku for STA type promotions)
		➦	promoprice (float) Promotional price
	➦	promosused (SubNode) Best promo price for used items	
		➦	promolinkid (string) Id of the promolink
		➦	promoid (string) Id of the parent promotion (or sku for STA type promotions)
		➦	promoprice (float) Promotional price

➦	productspecificpromotion (Node) List the STA type promotion associated with the product	
	➦	promolinkid (string) Id of the promolink
	➦	promoid (string) Id of the parent promotion (or sku for STA type promotions)
	➦	productstate (integer) ☰ 0 = All ☰ 1 = New ☰ 2 = Used
	➦	trigger (integer) Event that trigger the promotion, always 0 – Sales for STA
	➦	action (integer) How the promotion will be applied – always 0 – Fixed Price for STA
	➦	actionvalue (text) Value to be used when applying the promotion, promo price for STA
	➦	requiredqty (integer) Number of items required to make the promotion applicable
	➦	startdate (timestamp) Date and time at which the promo will be applicable
	➦	enddate (timestamp) Date and time the promotion is no longer applicable

➦	upccodes (Node)	
	➦	upcclassid (string) Id of the upc code's class (type)
	➦	upcvalue (string) Value of the UPC Code – can be used to search for the product

➦	inventoryqty (Node) Returns quantities for the target store or all the stores linked to the master

			qtynew (float)
			Total quantity of new items.
		➤	qtyused (float)
			Total quantity of used items.
		➤	qtyonlnew (float)
			Total quantity of new items available online.
		➤	qtyonlused (float)
			Total quantity of used items available online.
		➤	qtyrentonlu (float)
			Total quantity of items flagged as rentonly.
➤	Product Attributes		
	Products can have multiples attributes		
		➤	name (string)
			The name of the attribute
		➤	namefr (string)
			The name of the attribute French if it exists
		➤	nameen (string)
			The name of the attribute English if it exists
		➤	namesp (string)
			The name of the attribute Spanish if it exists
		➤	inputtype (integer)
			The type of attribute
			0 = Manual input
			1 = Single selection from a list
		➤	id (string)
			The identifier of the attribute
		➤	classid (string)
			The id of the parent attribute
		➤	sellable (boolean)
			Indicate if the attribute can be sold **(See Section 5.1)**
		➤	mandatorysell (boolean)
			Indicates if the attribute Must be sold with the product
		➤	applicationzone (string)
			Indicates the zone for which the attribute can be sold
		➤	productstate (integer)
			Some sellable attributes might only be sellable for certain state of product
			0 = Both New and Used
			1 = Only New
			2 = Only Used
		➤	value (string)
			The attribute value
		➤	valueid (string)
			The id of the value, for single selection inputtype
		➤	valuename (string)
			Name of the attribute's value, for single selection inputtype
		➤	valuenamefr (string)
			Name of the attribute's value, for single selection inputtype French if it exists
		➤	valuenameen (string)
			Name of the attribute's value, for single selection inputtype English if it exists
		➤	valuenamesp (string)
			Name of the attribute's value, for single selection inputtype Spanish if itexists

		valueprice (string) Value of the attribute's value , for single selection inputtype

Example:

```xml
<response>
 <request>
        <type>Product</type>
        <filters>
              <filter>
                    <field>currency</field>
                    <like>cad</like>
              <filter>
              <filter>
                    <field>sku</field>
                    <like>pric_132</like>
              <filter>
              <filter>
                    <field>productfilter</field>
                    <param>1</param>
              <filter>
        </filters>
 </request>
```

```xml
<count>1</count>
   <products>
        <product>
            <whenmod>1558653924.8301</whenmod>
            <sku>pric_132</sku>
            <alicecode>ABD#123456</alicecode>
            <name>Test Attribute</name>
            </desc>
            </release>
            <ismatrixparent>f</ismatrixparent>
            <parentsku>pric_32</parentsku>
            <released>t</released>
            <enable>t</enable>
            <taxable>t</taxable>
            <online>1</online>
            <sendtoecwid>f</sendtoecwid>
            <sendtopac>f</sendtopac>
            </prodseo>
            <buynew>5</buynew>
            <buyused>0</buyused>
            <sellnew>10</sellnew>
            <sellused>5</sellused>
            <size>2-4-5</size>
            <weight>5</weight>
            <platform>Batteries</platform>
            <category>Batteries</category>
            <categoryid>class_27m</categoryid>
            <upc>012345678912,TS-001</upc>
            <picture>atta_25</picture>
            <picturewhenmod>1480961159.46675</picturewhenmod>
            <variants> NODE
            <promoprice> NODE
            <upccodes>
                    <upcocde>
                            <upcclassid>class_123</upcclassid>
                            <upcvalue>012345678912</upcvalue>
                    </upccode>
                    <upccode>
                            <upcclassid>class_134</upcclassid>
                            <upcvalue>TS-001</upcvalue>
                    </upccode>
            </upccodes>
            <attributes>
                    <attribute>
                            <name>Eco Frais Qc</name>
                            <inputype>1</inputype>
                            <id>info_4705</id>
                            <classid>class_96</classid>
                            <sellable>t</sellable>
                            <mandatorysell>t</mandatorysell>
                            <applicationzoneid>class_100</applicationzoneid>
                            <productstate>0</productstate>
                            <value>TV 21 pouce</value>
                            <valueid>class_98</valueid>
                            <valuename>TV 21 pouce</valuename>
                            <valueprice>5</valueprice>
                    </attribute>
            </attributtes>
        </product>
        ...
   </products>
</response>
```

3.4 Images

Alice support multiple images for products, categories and multiple logos. For legacy purposes we kept the old single image request which returns the image with display order = 0.

3.4.1 Single Image
Get the primary image

Retrieves the primary image of the target (product, category, logo)

Parameters

☑	type: Image (string)
☐	sku (string) Only product images matching the SKU pattern will be returned (Sku can be the company id (comp_1) for logos or categroy id (class_xxx)
☐	name (string) Only product images matching the name pattern will be returned

Response

➤	id (string) The unique id of the image
➤	sku (string) The unique id of the image owner
➤	name (string) The name of the image file
➤	desc (string) Description of the image entered in Alice
➤	filesize (string) Size of the image
➤	whenmod (timestamp) Timestamp of the last image modification
➤	imageseo (string) SEO information defined for the image in Alice
➤	url (string) The URL of the product's image

Example:

```xml
<response>
 <request>
      <type>Image</type>
      <filters>
            <filter>
                  <field>sku</field>
                  <like>pric_1001</like>
            <filter>
      </filters>
 </request>
 <count>1</count>
 <products>
      <product>
            <id>atta_23</id>
            <sku>pric_1001</sku>
            <name>AG2003.jpg</name>
            </desc>
            <filesize>pric_10012m</filesize>
            <whenmod>2018-08-15 09:35:54.45178-04</whenmod>
            </imageseo>
            <url>http://website.com/path/1001_23.jpg</url>
      </product>
      ...
 </products>
</response>
```

3.4.2 Multiple Images

Get all of the images associated with the target

Retrieves all of the images associated to the target, they are group by owner for clarity where retrieving images for multiple owners.

Parameters

☑	type: MultiImage (string)
☐	sku (string) Only product images matching the SKU pattern will be returned (Sku can be the company id (comp_1) for logos or categroy id (class_xxx)
☐	name (string) Only product images matching the name pattern will be returned

Response

➡	imagecount (integer) Number of images associated with the owner
➡	sku (string) The unique id of the image owner
➡	images (node) List of images associated with the owner
	➡ id (string) The unique id of the image ➡ name (string) The name of the image file ➡ displayorder (integer) Order the images should be displayed in (0 based) ➡ desc (string) Description of the image entered in Alice ➡ filesize (string) Size of the image ➡ whenmod (timestamp) Timestamp of the last image modification ➡ imageseo (string) SEO information defined for the image in Alice ➡ url (string) The URL of the product's image

Example:

```xml
<response>
 <request>
        <type>MultiImage</type>
        <filters>
             <filter>
                  <field>sku</field>
                  <like>pric_1001</like>
             <filter>
        </filters>
 </request>
 <count>2</count>
 <products>
        <product>
             <imagecount>2</imagecount>
             <sku>pric_1001</sku>
             <images>
                 <image>
                    <id>atta_23</id>
                    <name>AG2003.jpg</name>
                    <displayorder>0</displayorder>
                    </desc>
                    <filesize>9699</filesize>
                    <whenmod>2018-08-15 09:35:54.45178-04</whenmod>
                    </imageseo>
                    <url>http://website.com/path/1001_23.jpg</url>
                 </image>
                 <image>
                    <id>atta_24</id>
                    <name>AG2006.jpg</name>
                    <displayorder>1</displayorder>
                    </desc>
                    <filesize>10042</filesize>
                    <whenmod>2019-07-16 13:21:45.5874-04</whenmod>
                    </imageseo>
                    <url>http://website.com/path/1001_24.jpg</url>
                 </image>
             </images>
        </product>
 </products>
</response>
```

3.5 Inventory

Get the various inventory values for products.

Returns the current number of items in inventory, this request can be limited to a single store or expanded to an entire network depending on access. The inventory numbers are ventilated.

Parameters

☑	type: Inventory (string)
☐	sku (string) Return the inventory count values for the product(s) using a SKU (pric_123)
☐	name (string) Return the inventory counts of the product matching the name pattern.
☐	group (string) Get product inventory for a specific group
☐	system (string) Get the inventory for a specific Alice system (or store)
☐	when (timestamp) Only products modified after the timestamp will be returned. ("when" and "whenonline" can also be used with the same result)

Response

➦	sku (string) The unique SKU of the product
➦	system (string) The ID of the system the inventory is for or ALL for a total count when not specified – since Alice manages multi-store networks
➦	group (string) The ID of the inventory group
➦	qtynew (integer) Number of new products
➦	qtyused (integer) Number of used products
➦	qtyonlnew (integer) Number of new products available online (excludes items on customer order)
➦	qtyonlused (integer) Number of used products available online (excludes items on customer order)
➦	qtyrentonly (integer) Number of products available for rent
➦	when (datetime) Tiemstamp of when the inventory was last updated for this product

Example:

```xml
<response>
 <request>
        <type>Inventory</type>
        <filters>
              <filter>
                    <field>sku</field>
                    <like>pric_10012m</like>
              <filter>
        </filters>
 </request>
 <products>
        <product>
              <sku>pric_10012m</sku>
              <system>ALL</system>
              <group>group_1</group>
              <qtynew>0</qtynew>
              <qtyused>1</qtyused>
              <qtyonlnew>0</qtyonlnew>
              <qtyonlused>0</qtyonlused>
              <qtyrentonly>0</qtyrentonly>
              <when>2015-02-17 13:35:35.271955-05</when>
        </product>
        ...
 </products>
</response>
```

3.6 Promotions
Get the promotion details and information.

Get the Alice promos and the list of products that are associated to the promotions. The response also lists the details about the triggers and actions to take when a promo needs to be applied.

There are 5 promotion actions in Alice, here is the way they affect the individual item price:
- Fixed Price: product price = actionvalue
- Discount Price: product price = price * (1 – (actionvalue/100))
- Bonus Percent: product buy price = price * (1 + (actionvalue/100)) (this is to buy items at an higher price for trade-ins)
- Cost Percent: product price = cost * (1 + (actionvalue/100))
- Cost Amount: product price = cost + actionvalue

Parameters
None

Response for promo node

	id (string) Identifier of the promo
	name (string) Name of the promo
	namefr (string) Name of the promo French if it exists
	nameen (string) Name of the promo English if it exists
	namesp (string) Name of the promo Spanish if it exists
	startdate (timestamp) Start Date and time of the promo
	enddate (timestamp) End date and time of the promo
	when (timestamp) Creation date of promo
	whenmod (timestamp) The time when the promo was last modified
	ownerid (string)
	Empty = Everybody classid = Contact type or company type (currently not sent to the API)
	promotype (integer)
	0 = All

	☰	1 = General
	☰	2 = Contact type (currently not sent to the API)
	☰	3 = Company type (currently not sent to the API)
	☰	4 = Membership (currently not sent to the API)
➦	online (integer)	
	Scope of the promos	
	☰	0 = Store only (currently not sent to the API)
	☰	1 = Web only
	☰	2 = Web and in store
➦	status (integer)	
	The status of the promo	
	☰	0 = Inactive
	☰	1 = Ongoing
	☰	2 = Upcoming
	☰	3 = Expired
➦	promolinks (see next section)	
	List of products that the promo applies to	

Example:

```xml
<promos>
 <promo>
   <id>promo_4</id>
   <name>Test Promo</name>
   <startdate>2019-05-24 08:00:00-04</startdate>
   <enddate/>
   <when>2019-05-24 11:57:13.416014-04</when>
   <whenmod>2019-05-24 11:57:13.416016-04</whenmod>
   <ownerid/>
   <promotype>1</when>
   <online>2</when>
   <status>1</status>
   <promolinks>
      *** SEE NEXT SECTION FOR PROMOLINKS NODE EXAMPLE ***
   </promolinks>
 </promo>
</promos>
```

Response for Promolink Child Node in Promo

↱	promolinkid (string) The ID of the promolink.
↱	promoid (string) The ID of the promo that the promolink is associated with.
↱	ownerid (string) The type of item the promo applies to.
	▤ pric_* = applies to a product ▤ class_* = applies to a category or services ▤ empty = applied to all the inventory ▤ plink_* = applies to promolinkid (in that case check the next node (productid) to see on what it is applied.
↱	productid (string)
↱	productstate (integer)
	▤ 0 = All ▤ 1 = New ▤ 2 = Used
↱	trigger (integer)
	▤ 0 = Sale ▤ 1 = SubPromo (plink_) If a SubPromo get the parent promo data (requiredqty, Ownerid, trigger, ProductState) ▤ 2 = Purchase
↱	action (integer)
	▤ 0 = Fixed price ▤ 3 = Discount price ▤ 4 = Bonus percent ▤ 5 = Cost percent ▤ 6 = Cost amount
↱	actionvalue (float) The value or percentage to apply depending on the action
↱	promocode (string) (Not currently supported)
↱	promocodeonly (boolean) (Not currently supported)
↱	requiredqty (integer) Required quantity for the promo to be applied
↱	allinventory (boolean) The action applies the complete inventory
↱	promocondition (boolean) If true, then needs to check the ownerid(promolinkid) to get the action to apply.
↱	promotype (integer)
	▤ 0 = All

	1 = General 2 = Contact type (currently not sent to the API) 3 = Company type (currently not sent to the API) 4 = Membership (currently not sent to the API)
	promoownerid (string) If the promo is associated with a contact type, company type or Membership (currently not sent to the API)

Example: (10% discount when pric_73 is sold)

```
<promos>
 <promo>
     *** SEE PREVIOUS SECTION FOR PROMO NODE EXAMPLE ***
   <promolinks>
       <promolinkid>plink_4</promolinkid>
       <promoid>promo_4</promoid>
       <ownerid>pric_73</ownerid>
       <productid/>
       <productstate>0</productstate>
       <trigger>0</trigger>
       <action>3</action>
       <actionvalue>10</actionvalue>
       <promocode/>
       <promocodeonly>f</promocodeonly>
       <requiredqty>1</requiredqty>
       <allinventory>f</allinventory>
       <promocondition>f</promocondition>
       <promotype>1</promotype>
       <promoownerid/>
   </promolinks>
 </promo>
</promos>
```

3.7 Tax
Get the tax details.

Getting tax details is useful for posting transactions using the API or to sync them with an online store.

Parameters
None

Response

➡	taxclassid (string) Unique tax ID from Alice
➡	taxname (string) The tax rate's name
➡	taxnamefr (string) The tax rate's name in French if it exists
➡	taxnameen (string) The tax rate's name in English if it exists
➡	taxnamesp (string) The tax rate's name in Spanish if it exists
➡	taxrate (float) The tax rate (5 = 5%)

Example:

```xml
<taxes>
 <tax>
   <taxclassid>class_12</taxclassid>
   <taxname>TVQ</taxname>
   <taxrate>9.975</ratxrate>
 </tax>
 <tax>
   <taxclassid>class_11</taxclassid>
   <taxname>TPS</taxname>
   <taxrate>5</taxrate>
 </tax>
</taxes>
```

3.8 System Application Zone

Get the system's application zone, for sellable attributes.

Required to determine if a sellable attribute must be added to the invoice.

Parameters

None

Response

➦	systemid (string) Id of the system to which the request was sent
➦	applicationzoneid (string) System application zone class id

Example:

```xml
<applicationzone>
 <zone>
   <systemid>sys_865</systemid>
   <applicationzoneid>class_100</applicationzoneid>
 </zone>
</applicationzone>
```

3.9 Application Zone List
Get the application zone list.

The list of all application zones on the system could be used to match it with a website customer's location for province/state/region specific sellable attribute (ie. electronics or tires environmental fees).

<u>Parameters</u>
None

<u>Response</u>

↱	zoneclassid (string) Unique application zone ID from Alice
↱	zonename (string) The application zone's name
↱	zonenamefr (string) The application zone's name in French if it exists
↱	zonenameen (string) The application zone's name in English if it exists
↱	zonenamesp (string) The application zone's name in Spanish if it exists

<u>Example:</u>

```
<applicationzone>
 <zone>
   <zoneclassid>class_100</zoneclassid>
   <zonename>Zone 1</zonename>
 </zone>
 <zone>
   <zoneclassid>class_101</zoneclassid>
   <zonename>Zone 2</zonename>
 </zone>
</applicationzone>
```

3.10 Accounting Posting (GLExportData)

Get the date for a general ledger writing.

Returns the raw data from the "Accounting Posting" report in Alice. This can be used to create a general ledger entry in a third party accounting application for the specified period. The field description includes the formula used for calculated value and if values are combined in the report (ie expectedinventoryend and stockcountinv(added/deleted)).

Parameters

☑	start: (date) – Format (YYYY-MM-DD) Day the period of the report begins
☑	end (date) – Format (YYYY-MM-DD) Day the period of the report ends

Request Example

```xml
<response>
 <request>
        <type>GLExportData</type>
        <filters>
             <filter>
                  <field>start</field>
                  <param>2019-06-01</param>
             <filter>
             <filter>
                  <field>end</field>
                  <param>2019-06-15</param>
             <filter>
        </filters>
```

Response

➡	tax collected (float) Total tax collected
➡	productsales (float) Total product sales
➡	servicesales (float) Total service sales
➡	attributesales (float) Total saleable attributes sales
➡	membershipsales (float) Total membership sales
➡	rentals (float) Total rental fees collected
➡	product refund (float) Total value of refunded product (amount refunded to customer)
➡	servicerefund (float) Total value of refunded services (amount refunded to customer)
➡	attributerefunds (float) Total value of refunded saleable attributes (amount refunded to customer)

↱	inventorystart (float) Value of the inventory at the beginning of the period	
↱	inventoryend (float) Value of the inventory at the end of the period	
↱	supplierpuchase (float) Amount purchased from suppliers	
↱	suppliershipping (float) Shipping paid on supplier purchases	
↱	tradeins (float) Value of traded in items (items exchanged by a customer as payment - added to inventory)	
↱	inventoryretrieval (float) Value of items received from a linked store in the same network (inter-store transfers)	
↱	cogs (float) – Calculated Value Cost Of Goods Sold - Deprecated Inventortstart + (stockcountinvadded – stockcountinvdeleted) + supplierpurchase + suppliershipping + tradeins + inventoryretrieval – inventoryend	
↱	cogsclean (float) – Calculated Value Cost Of Goods Sold – Value used in the report productsalescost + producttransfercost	
↱	giftcardissued (float) Value of giftcard issued to customer - sold in invoices	
↱	giftcardtaken (float) Value of giftcard used as payment in invoices	
↱	manualgiftcardissued (float) Value of giftcard manually created in the system – not sold in invoices as giftcards	
↱	manualgiftcardtaken (float) Value of giftcard reduced manually with the system's tool - not used as payment	
↱	issuedcredit (float) Store credit given to customer as refund	
↱	takencredit (float) – Not displayed, used in calculations Store credit used by customer as payment	
↱	finalizedtakendeposit (float) – Not displayed, used in calculations Deposit on order/repair that were finalized during the period	
↱	issueddeposit (float) Deposit added to an order/repair during the period	
↱	paymodetakendeposit (float) Deposit on order/repair that are have been refunded	
↱	manualcreditissued (float) Store credit increased manually	
↱	manulcredittaken (float) Store credit reduced manually	
↱	roundedamount (float) Sum of all rounding of cash payments by the penny rounding feature	
↱	takenchargetoaccount (float) Payments made to a customer's creditline	
↱	issuedchargetoaccount (float) Invoice charged to a customer's creditline	
↱	takendebitfromaccount (float) Payment made to a customer's creditline with a credited invoice	
↱	issueddebitfromaccount (float) Invoice credited to a customer's creditline	

	consignementpayment (float) Amount paid to customer's for their sold consignments
	stockcountinvadded (float) Value of items added to inventory during stockcount(s)
	stockcountinvdeleted (float) Value of items removed from inventory during stockcount(s)
	deletedinventory (float) Total value of items deleted from inventory (includes stockcountinvdeleted) Display Line "Deleted Inventory" = deletedinventory - stockcountinvdeleted
	customservice (float) Total service sold that will be included in the calculation of the custom margin
	cashback (float) Sum of payment made to customer (winning lottery tickets)
	customservicerefunds (float) Total service refunded that will be included in the calculation of the custom margin
	depositstart (float) Total value deposit at the beginning of the period
	depositend (float) Total value deposit at the end of the period
	giftcardstart (float) Total value of giftcards at the beginning of the period
	giftcardend (float) Total value of giftcards at the end of the period
	creditstart (float) Total value of store credit at the beginning of the period
	creditend (float) Total value of store credit at the end of the period
	receivablestart (float) Total value of account receivable (creditline) at the beginning of the period
	receivableend (float) Total value of account receivable (creditline) at the end of the period
	incomes (float) – Calculated Value Total gross income productsales + servicesales + attributesales + membershipsales + rentals – cashback – productrefund – servicerefund - attributerefunds
	grossprofit (float) – Calculated Value Gross profit margin in currency – Deprecated incomes – cogs – servicecost
	grossprofitpercent (float) – Calculated Value Gross profit margin, as percentage – Deprecated (grossprofit/incomes) x 100
	grossprofitclean (float) – Calculated Value Gross profit margin in currency incomes – cogsclean – servicecost
	grossprofitpercentclean (float) – Calculated Value Gross profit margin as percentage (grossprofitclean/incomes) x 100
	customincomes (float) – Calculated Value Total gross income, excludes services flagged as "excluded from gross margin" in Alice productsales + customservice + attributesales + membershipsales + rentals – cashback – productrefund – customservicerefunds – attributerefunds
	customgrossprofit (float) – Calculated Value

	Gross profit margin in currency, excludes services flagged as "excluded from gross margin" in Alice – Deprecated customincomes – cogs – customservicecost
	customgrossprofitpercent (float) – Calculated Value Gross profit margin, as percentage, excludes services flagged as "excluded from gross margin" in Alice – Deprecated (customgrossprofit/customincomes) x 100
	customgrossprofitclean (float) – Calculated Value Gross profit margin in currency, excludes services flagged as "excluded from gross margin" in Alice customincomes – cogsclean – customservicecost
	customgrossprofitpercentclean (float) – Calculated Value Gross profit margin as percentage, excludes services flagged as "excluded from gross margin" in Alice (customgrossprofitclean/customincomes) x 100
	takendeposit (float) – Calculated Value Total taken deposit Used to display line "Taken Order Deposit" finalizedtakendeposit + paymodetakendeposit
	servicecost (float) Cost of services sold
	customservicecost (float) Cost of services sold, excluded services flagged as "excluded from gross margin" in Alice
	taxpaid (float) Taxes paid on supplier purchase
	expectedinventoryend (float) – Calculated Value Expected value of the inventory at the end of the period, based on start inventory and other inventory variations. Used in turn to calculate inventory movement inventorystart + (inventorycreated – stockcountinvadded) + costadjustment + supplierpurchase + tradeins + inventoryretreival + (stockcountinvadded – stockcountinvdeleted) + productrefundcost – cogsclean – (deletedinventory – stockcountinvdeleted)
	costadjustment (float) Inventory variation caused by manual cost adjustments or pricechanges
	inventorycreated (float) – Used in calculation only Value of inventory added not with a purchase (refund, tradeins and consignment)
	producttransfersales (float) Value of products transferred to a linked store in the same network (inter-store transfer)
	producttransfercost (float) Cost of products transferred to a linked store in the same network (inter-store transfer) Used to calculate the COGS
	productsalescost (float) Cost of product sold
	canceledsalesadjust (float) Adjustment for canceled sales – Deprecated
	canceledpurchaseadjust (float) Adjustment for canceled purchases - Deprecated
	canceledrefundadjust (float) Adjustment for canceled refunds – Deprecated
	inventorymovement (float)

		Delta between the expected and calculated inventory at the end of the period based on the calculated stats generated at the end of each day		
➦		canceledissuedcredit (float) Value of store credit issued in an invoice that was canceled during the period		
➦		canceledtakencredit (float) Value of store credit taken as payment in an invoice that was canceled during the period		
➦		canceledpreviouscredit (float) Value of store credit issued or taken outside the period but for which the invoice was canceled during the period Used to adjust the value of cancelissuedcredit or canceledtakencredit based on the delta		
➦		prepaidcardissued (float) Value of prepaid cards issued during the period		
➦		canceledissuedgiftcard (float) – Deprecated		
➦		canceledtakengiftcard (float) – Deprecated		
➦		canceledpreviousgiftcard (float) – Deprecated		
➦		Paymodedata (Node)		
	➦	paydetails (sub-node)		
		➦	paymode (string) Name of the payment mode as displayed in Alice	
		➦	paymodeissueddeposit (float) Amount of deposit made with the payment mode (included in issueddeposit)	
		➦	paymodetakendeposit (float) Amount of deposit refunded during the period with the payment mode (included in paymodetakendeposit and finalizedtakendeposit)	
		➦	paymodeissuedcredit (float) Amount of credit issued with the payment mode, only for credit payment mode (included in issued credit)	
		➦	paymodetakencredit (float) Amount of credit taken as payment with the payment mode, only for credit payment mode (included in issued credit)	
		➦	paymodetotal (float) Total amount collected for that payment mode	
➦		Taxdata (Node)		
	➦	taxdetails (sub-node)		
		➦	taxname (string) Name of the tax as displayed in Alice (and returned by the Tax Request)	
		➦	taxtotal (float) Amount collected for that tax (included in taxcollected)	
➦		ventilateddata (Node) Sales/Refund/Cost totals ventilated by account number		
	➦	accounts (sub-node)		
		➦	accountnumber (string) Account number assigned to the products or categories	
		➦	productsales (float) Total product sales	
		➦	productrefund (float) Total product refunds	
		➦	productrefundcost (float) Total cost of refunded products	

			➡	productcost (float) Total cost of sold	
			➡	servicesales (string) Total service sales	
			➡	servicerefund (float) Total service refunds	
			➡	servicecost (float) Total cost of service sold	
			➡	customservicesales (string) Total service sales (the ones not excluded from gross margin)	
			➡	customservicerefunds (float) Total service refunds (the ones not excluded from gross margin)	
			➡	customservicecost (float) Total cost of service sold (the ones not excluded from gross margin)	

3.11 Services

Get Alice's services

This returns all the services in the store. Services are a special type of products in Alice which do not have inventory. Examples are protection services, fishing licenses, repair fees and room rentals.

Parameters
None

Response

➡	serviceid (string) Id to be used a the service's sku when it is added to an order
➡	servicename (string) Name of the service
➡	servicenamefr (string) Name of the service in French if it exists
➡	servicenameen (string) Name of the service in English if it exists
➡	servicenamesp (string) Name of the service in Spanish if it exists
➡	servicetype (integer) Type of the service
	☰ 8 = General ☰ 18 = Repair – applicable on repair invoices ☰ 19 = Protection – aka "Service plan"
➡	serviceprice (float) Retail price of the service
➡	taxexempt (bool) Indicates that the service is taxable or not
➡	picture (string) Id fo the image linked to the service
➡	servicecatid (string) Id of the service's category
➡	servicecatname (string) Name of the service's category
➡	protectionrange (node) Protection services prices can be set based on the price of the product on which they are applied.

	➡ range
	   ➡ rangefrom (float)
	     Bottom end of the price range
	     ➡ rangeto (float
	     Upper end of the price range

			↪	price (float) Retail price of the protection service if the related product's price is in the range.

3.12 Shipping Size and Weight (ShippingAttributes)
Get the classid and name of the shipping attributes (size and weight)

The dimension and weight of the product can be used to determine exact shipping cost using delivery services plugins. This request retrieves the attribute's detailsthat were configured in Alice by the store owner. Specific product size and weight are returned on new fields in the product response and included in the product's attributes list.

Parameters

None

Response

↱	sizeclassid (string)
	Unique Id of the size attribute
↱	sizename (string)
	Name given to the size attribute in the store's system
↱	weightclassid (string)
	Unique Id of the weight attribute
↱	weightname (string)
	Name given to the weight attribute in the store's system

Example:

```xml
<shippingattributes>
    <details>
        <sizeclassid>class_112</sizeclassid>
        <sizename>Dimension (W-L-H)</sizename>
        <weightclassid>class_114</weightclassid>
        <weightname>Weight (lbs)</weightname>
    </details>
</shippingattributes>
```

3.13 UPCList
Get the classid and name of all active UPC Code Types

Alice supports multiple codes for the same product, while they are called UPC for simplicity, they can be in any format, this returns the list of all the available UPC Type and their names.

<u>Parameters</u>
None

<u>Response</u>

➡️	upcclassid (string) Unique Id of the upc code type
➡️	upcname (string) Name given to the upc code type in the store's system
➡️	upcnamesp (string) Name given to the upc code type in the store's system (French)
➡️	upcnameen (string) Name given to the upc code type in the store's system (English)
➡️	upcnamesp (string) Name given to the upc code type in the store's system (Spanish)

<u>Example:</u>

```
<upclist>
    <details>
        <upclassid>class_112</upcclassid>
        <upcname>Supplier Code</upcname>
    </details>
    <details>
        <upclassid>class_1354</upcclassid>
        <upcname>UPC #1</upcname>
    </details>
</upclist>
```

3.14 Entity List
Get the list of contacts and companies in Alice

Request that returns the name, creditline values and contact information of the clients. In Alice we separate them in contact for individuals and companies for legal entities.

Parameters

☐	mode (integer) --- PARAM The currency used for the pricing fields. Products can have different prices in different countries.

	▤	0 = All – Contacts and Companies
		1 = Contacts only
		2 = Companies only

☐	lastsync (datetime) – Format (YYYY-MM-DD HH:MM:SS) --- PARAM Date time from which the data will be grabbed. To ge only modified contacts

Response

➦	entityid (string) The unique id of the client
➦	entitymode (string) Indicates if it's a contact or company
➦	entityname (string) Name of the client
➦	entitytypeid (string) Id of the client's type, can be used for certain promotions.
➦	entityphone (string) Client's phone number, can be empty. (123) 456-7890
➦	entityemail (string) Client's email, can be empty.
➦	entitywebsite (string) Client's website, can be empty
➦	entitylongaddress (string) Client's address, long format (street, city, state, country, postal code)
➦	entityshortaddress (string) Client's street address.
➦	entitycreditbalance (float) Client's current available credit.
➦	entitycreditlimit (float) Client's credit line limit.

Example:

```xml
<response>
 <request>
       <type>EntityList</type>
       <filters>
             <filter>
                   <field>mode</field>
                   <param>2</param>
             <filter>
       </filters>
 </request>
 <entitylist>
       <entity>
             <entityid>cont_1</entityid>
             <entitymode>contact</entitymode>
             <entityname>John Doe</entityname>
             <entitytypeid>class_1m</entitytypeid>
             <entityphone>(123) 555-1234</entityphone>
             <entityemail>j.doe@test.ca</entityemail>
             <entitywebsite/>
             <entitylongaddress>123 Main st, Sometown, Somestate,
H0H0H0</entitylongaddress>
             <entityshortaddress>123 Main st</entityshortaddress>
             <creditlinebalance>1988.45</creditlinebalance>
             <creditlinelimit>2500</creditlinelimit>
       </entity>
       ...
 </entitylist>
</response>
```

3.15 Promotion Ended
Get the Alice codes for all products for which a promotion has ended

Request that returns an array containing the Alice codes for all products for which a promotion has ended since the timestamp passed as parameter.

Parameters

☑	start (datetime) – Format (YYYY-MM-DD HH:MM:SS) --- PARAM Date time from which the data will be grabbed. To ge only modified contacts

Response

➦	promoendednew->alicecodes (array of string) Alice codes of the products for which a promotion on the NEW state has ended
➦	promoendedused->alicecodes (array of string) Alice codes of the products for which a promotion on the USED state has ended

Example:

```xml
<response>
 <request>
      <type>EntityList</type>
      <filters>
           <filter>
                <field>mode</field>
                <param>2</param>
           <filter>
      </filters>
 </request>
 <data>
      <promoendednew>
           {PRE#123456,PRE#654678}
      </promoendednew>
<data>
</data>
      <promoendedused>
           {PRE#784512,PRE#123456}
      </promoendedused>
 </data>
</response>
```

4.1 Transaction

Transaction/order creation.

Creates transaction in Alice. Also creates a customer if it doesn't already exist.

IMPORTANT: This request writes data into the store's database. Do not perform test on a production version of the store.

IMPORTANT: This request must only be sent to a store, never a master or submaster.

IMPORTANT: Online transactions require a specific configuration in Alice POS, these can be found in the "Web Site" tab of the Alice settings menu. Consult https://support.alicepos.com/en/knowledge-base/how-to-link-alice-pos-to-your-online-store/ for more details.

Parameters

☑	type: Transaction_Add (string)					
☑	scheme Order scheme					
	☑	order Order information				
		☐	id (string) Unique order ID from an online store			
		☑	currency (string) Order currency			
			▤	CAD = Canadian Dollars		
			▤	USD = United States Dollars		
			▤	USDZ1 = United States (zone 1) Dollars		
			▤	COP = Colombian Pesos		
			▤	EGP = Egyptian Pounds		
		☑	discount_total (float) Discount applied to the order			
		☑	shipping_total (float) Shipping total **without tax**			
		☑	shipping_tax (float) Shipping tax amount			
		☑	total (float) Order total **including tax**			
		☑	total_tax (float) Total tax amount **including shipping tax**			
		☑	customer_id (string) Unique customer ID from an online store – used to match the online customer to an Alice customer.			

			☑	shipto_id (string) Unique ship to ID from an online store – used to match the online customer to an Alice customer. (Even if it's the same as the customer_id)		
			☑	taxes List of taxes applied to the order (**excluding shipping tax**)		
				☑	tax	
					☑	class (string) Tax class (can be found using Tax at 3.7)
					☑	rate (float) Rate associated with the class (5.0 for 5%)
					☑	amount (float) Tax amount applied
			☐	alice_client_id (string) Unique customer ID from Alice (if it exists)		
			☐	reference (string) Invoice reference number, can be used to indicate the online order #		
			☐	invoicenote (string) Note the will be added to the invoice, printed on the receipt		
			☐	internalnote (string) Note the will be added to the invoice, only visible in Alice.		
		☑	billing Billing information			
			☑	first_name (string) Customer's first name		
			☑	last_name (string) Customer's last name		
			☐	company (string) Billing company		
			☑	address_1 (string) Billing address 1		
			☐	address_2 (string) Billing address 2		
			☑	city (string) Billing city		
			☑	state (string) Billing state or province		
			☑	postcode (string) Billing zip or postal code		
			☑	country (string) Billing country		
			☑	email (string) Customer's email address		
			☑	phone (string) Customer's phone number		
			☑	type (string) Determine if the client will be saved as a 'contact' or 'company' in Alice. Only for new entries, if one already exists with the ecomid, it will not be changed.		
		☑	shipping Shipping information			
			☑	first_name (string) Customer's first name (shipping)		

			☑	last_name (string) Customer's last name (shipping)				
			☐	company (string) Shipping company				
			☑	address_1 (string) Shipping address 1				
			☐	address_2 (string) Shipping address 2				
			☑	city (string) Shipping city				
			☑	state (string) Shipping state or province				
			☑	postcode (string) Shipping zip or postal code				
			☑	country (string) Shipping country				
			☑	type (string) Determine if the client will be saved as a 'contact' or 'company' in Alice. Only for new entries, if one already exists with the ecomid, it will not be changed.				
			☐	taxes List of taxes applied to the shipping				
				☑	tax			
					☑	class (string) Tax class (can be found using Tax at 3.7)		
					☑	rate (float) Rate associated with the class (5.0 for 5%)		
					☑	amount (float) Tax amount applied		
	☑	products List of products ordered						
		☑	product					
			☐	product_id (string) Unique product ID from an online store				
			☑	quantity (int) Quantity ordered				
			☑	total (float) <mark>Price at which the product was sold (unit price) (without tax)</mark>				
			☑	total_tax (float) Total tax amount applied to the product				
			☑	sku (string) The unique SKU of the product in Alice – 'pric_XXX' Or service classid = 'class_XXX' **(See Section 5.1 for Sellable Attributes)**				
			☑	price (float) <mark>The base product unit price (default retail price)</mark> **(See Section 5.1 for Sellable Attributes)**				
			☑	used (boolean) If the product is new (false) or used (true)				
			☐	note (string) Add note that will be displayed beside the product on the invoice				

					☐	promolinkid (string) Id of the promotion element from the product->promoprice node.		
					☐	promoid (string) Id of the promotion from the product->promoprice node.		
					☐	taxes		
						☑ tax		
							☑ class (string) Tax class (can be found using Tax at 3.7)	
							☑ rate (float) Rate associated with the class (5.0 for 5%)	
							☑ amount (float) Tax amount applied	

Response

➦	webid (string) Unique order ID given in the request			
➦	aliceid (string) The unique ID given to the order by Alice			
➦	client The customer			
	➦	webid (string) Unique customer ID given in the request (auto generated if empty)		
	➦	aliceid (string) The unique ID given to the customer by Alice		
	➦	operation (string) If the customer was created or updated (create/update)		
➦	products Ordered products list			
	➦	product Ordered product		
		➦	webid (string) Unique product ID given in the request	
		➦	aliceid (string) The product SKU (unique ID from Alice)	
		➦	inventory Product stock	
			➦	quantity_new (int) Quantity in stock (new)
			➦	quantity_used (int) Quantity in stock (used)
			➦	qty_available_new (int) Quantity available for order (new)
			➦	qty_available_used (int) Quantity available for order (used)
		➦	price Product prices	
			➦	regular_price (float) The product's regular price
			➦	used_price (float) The product's regular price (used)
➦	operation (string) If the order was created (create)			

Example:

```xml
<response>
 <request>
  <credentials></credentials>
  <scheme></scheme>
 </request>
*** FULL REQUEST WILL BE INCLUDED,SEE PREVIOUS SECTION FOR DETAILS ***
 <return>
  <order>
   <webid>105523</webid>
   <aliceid>C190524-2</aliceid>
   <client>
     <webid>ecom66</webid>
     <aliceid>cont_3659</aliceid>
     <operation>update</operation>
   </client>
   <products>
     <product>
       <webid>1783</webid>
       <aliceid>pric_103</aliceid>
       <inventory>
         <quantity_new>3</quantity_new>
         <quantity_used>0</quantity_used>
         <qty_available_new>1</qty_available_new>
         <qty_available_used>0</qty_available_used>
       </inventory>
       <price>
         <regular_price>199.95</regular_price>
         <used_price>0</used_price>
       </price>
     </product>
   </products>
   <operation>create</operation>
  </order>
 </return>
 <status>
   <code>200</code>
   <description>Success</description>
 </status>
</reponse>
```

5.1 Sellable Product Attributes

Some product attributes can be sold in Alice and Online, this can be optional or mandatory and depends on the system's and the attribute's application zone.

Practical applications for this are environmental fees on electronics/tires, warranties and mandatory deposits on recyclable drink containers.

All the necessary fields are returned in the attributes section of the product request (section 3.3).

Application Zone

This defines area in which a sellable attribute can/must be sold. The list of all application zone for the system and the system's own applications (usually its location) can be retrieve using the request explained in section 3.8 and 3.9

The decision to use the online customer's or the system's zone will be made the store owner. It varies from jurisdictions to jurisdictions.

Example: Quebec's environmental fees must be charged on all electronics bought at a store located in Quebec or shipped to a customer living in Quebec but are not applicable in any other cases.

Mandatory Sell

Some attributes must be sold when the product they are attached to is sold, the quantities must match. As explained above, the application zone must be checked.

ID and Price

The values to use for the "sku" and "price" fields in the transaction request are different than regular products and depend on the attribute's inputtype.

For manual values (inputtype = 0), the "sku" will be the attribute's classid and "price" will be the attribute's value.

For single selection (inputtype =1), the "sku" will be the attribute's valueid and the "price" will be the attribute's valueprice.

In both cases the attribute's name will be used on the invoice.

5.2 Product Matrix

It is possible to configure product matrix in Alice POS, that information is also returnable by the API in a few different ways.

Both the matrix parent and the variations are individual skus in Alice. They can have their own image, UPC, prices, promos, etc.

Up to 3 different attributes can used to create variations with any number of possible values for each of those attributes.

The *ismatrixparent* field indicates that the product has variations. If it does, the *variants* node will be present and contain all the variations details (list of variantion ids, list of possible attributes (name, id, value names, value ids).

The information in the *variants* node is designed to allow integrators to generate all of the product's options without having to load/create all the sub-products.

Variations are name using this format: "Parent's name [Attribute 1 Value Name – Attribute 2 Value Name – Attribute 3 Value Name]". Ie "*Matrix Shirt [Large – Green]*". They also have a value in the *parentsku* field in the product response.

When generating an order, the specific variation's details must be used in the product node, not the parent's.