

# API and integrations

Raley PO exposes HTTP REST endpoints which you can use to facilitate B2B integration with your own system(s).

## Authorization

The authorization is implemented via HTTP header "Authorization". Every time you execute a method call on our API you have to provide a valid value in this header.

To get the value of Authorization header, navigate to Raley Apps configuration page and then click on API & Integrations tab. Then copy the value from API Key and use in your application(s) to retrieve and modify data in Raley PO.

You can always regenerate the Authorization header, however, make sure that you've updated your systems which are making use of it.

## Available APIs

The following table lists available REST API methods that a third party service can invoke. URL's must start with <https://po.raleyapps.com/> and end with the URL suffix as indicated in the table below

Method	URL suffix	Description	Example Response
GET	<code>api/v1/external/get-suppliers</code>	Returns list of suppliers configured for given installation of Raley PO. Archived suppliers are not returned	<pre>[ {   "id": 291,   "name": "Test supplier",   "phone": "555 5555",   "email": "contact@supplier.com",   "address": "Address of supplier",   "notes": "",   "paymentDetails": "DE72392732 643627",   "taxNo": "DE8473273",   "contact": "",   "terms": "NET30",   "archived": false } ]</pre>
GET	<code>api/v1/external/get-taxes</code>	Returns list of taxes configured for given installation of Raley PO. Archived taxes are not returned	<pre>[ {   "id": 120,   "name": "Default tax",   "value": "20.0",   "isDefault": true,   "archived": false,   "companyId": 217 } ]</pre>
GET	<code>api/v1/external/find-products-by-sku?sku=SKU_VALUE</code>	<p>Finds and returns list of product that match by given SKU identifier SKU_VALUE. Both, active and archived products can be returned.</p> <p>This methods returns HTTP 404 if no products are found.</p>	<pre>[ {   "id": 1413,   "sku": "abc",   "description": "Mazutka",   "unitPrice": 5323.0,   "supplierId": 291,   "taxRate": {     "id": 120,     "name": "Default tax",     "value": "20.0",     "isDefault": true,     "archived": false,     "companyId": 217   },   "active": true,   "approved": true,   "productApprovalType": null } ]</pre>

POST	<code>api/v1/external/create-product</code>	<p>Creates a product with given parameters passed as JSON array.</p> <p>sku - business identifier used to identify products in your system</p> <p>unitPrice - product price including applicable taxRate</p> <p>supplierId - id of supplier that offers this product</p> <p>taxRate - object containing an "id" field which corresponds to applicable tax rate</p> <p>active - whether the product should be active or archived</p> <p>approved - true when you're not using product approval process. Otherwise false.</p> <p>productApprovalType - null if you're not using the product approval process, otherwise a JSON object with mandatory "id" field which corresponds to the product approval type ID.</p> <p>On successful invocation, the method will return newly created product data in JSON format.</p>	<pre>{   "sku": "YOUR BUSINESS IDENTIFIER",   "description": "This is a brand new product that we add to Raley",   "unitPrice": 199.99,   "supplierId": 291,   "taxRate": {     "id": 120   },   "active": true,   "approved": true,   "productApprovalType": null }</pre>
PUT	<code>api/v1/external/update-product</code>	<p>Updates a product with given id to the values set in product fields.</p> <p>If the product specified by id is not found then this method will return HTTP 404</p> <p>The semantic of the fields is the same as in create-product method.</p> <p>On successful invocation, the method will return updated product data in JSON format</p>	<pre>{   "id": 1733,   "sku": "YOUR BUSINESS IDENTIFIER 2",   "description": "This is a brand new product that we add to Raley 22",   "unitPrice": 199.99,   "supplierId": 291,   "taxRate": {     "id": 120   },   "active": true,   "approved": true,   "productApprovalType": null }</pre>
POST	<code>api/v1/external/create-supplier</code>	<p>Creates a supplier with specified fields. On successful invocation this method return created supplier in JSON format</p>	<pre>{   "name": "My supplier",   "phone": "555 55 55",   "email": "info@mysupplier.com",   "address": "My supplier's address",   "notes": "Any notes you want about this supplier",   "paymentDetails": "NET30",   "taxNo": "885732345838",   "contact": "Jack Rabbit",   "terms": "Any terms you deem important here" }</pre>
PUT	<code>api/v1/external/update-supplier</code>	<p>Updates supplier with specified fields.</p> <p>If the supplier specified by id was not found then this method returns HTTP 404</p> <p>On successful invocation this method return updated supplier in JSON format</p>	<pre>{   "id": 1386,   "name": "My supplier A",   "phone": "555 55 55",   "email": "info@mysupplier.com",   "address": "My supplier's address",   "notes": "Any notes you want about this supplier",   "paymentDetails": "NET30",   "taxNo": "885732345838",   "contact": "Jack Rabbit",   "terms": "Any terms you deem important here" }</pre>