

Price2Spy API SOAP Documentation



Authors:

Miša Krunic, misha@price2spy.com,

Nemanja Đokić n.djokic@price2spy.com

Version: 2.10

Table of contents

Introduction.....	2
List of operations.....	2
Operations definition	3
Matching Suggestions	35
Introduction.....	35
List of operations.....	35
Operations definition	35
Implementation of Price2Spy API Clients - SOAP	38
PHP	38
PHP code samples	38
In case of HTTPS-related connection problems	38

Introduction

Price2Spy API SOAP document describes details of Price2Spy API operations – implemented in SOAP.

Before you read this document, please make sure you have read and understood **Price2Spy API Reference Guide** document.

List of operations

- Managing Price2Spy data (insert / update / delete operations)
 - i. **insertProduct**
 - ii. **updateProduct**
 - iii. **deleteProduct**
 - iv. **insertUrl**
 - v. **updateUrl**
 - vi. **deleteUrl**
 - vii. **insertCategory**
 - viii. **updateCategory**
 - ix. **deleteCategory**
 - x. **insertBrand**
 - xi. **updateBrand**
 - xii. **deleteBrand**
 - xiii. **insertSupplier**
 - xiv. **updateSupplier**
 - xv. **deleteSupplier**
 - xvi. **approveMatchingSuggestions**
 - xvii. **rejectMatchingSuggestions**
 - xviii. **insertMetadata1**
 - xix. **updateMetadata1**
 - xx. **deleteMetadata1**

- xxi. **insertMetadata2**
- xxii. **updateMetadata2**
- xxiii. **deleteMetadata2**
- xxiv. **insertMetadata3**
- xxv. **updateMetadata3**
- xxvi. **deleteMetadata3**
- Retrieving Price2Spy data ('get' operations)
 - xxvii. **getProducts** – get products for given search criteria
 - xxviii. **getUrls** – get URLs for given product(s)
 - xxix. **getMeasurements** – get price checks (measurements) for given URL
 - xxx. **getCurrentPricingData** – get all Price2Spy data for products matching given search criteria
 - xxxi. **getBrands** – get all brands
 - xxxii. **getCategories** – get all categories
 - xxxiii. **getSuppliers** – get all suppliers
 - xxxiv. **getPriceBuckets** – get all price buckets
 - xxxv. **getMatchingSuggestions** – get Matching Suggestions for products fulfilling given search criteria
 - xxxvi. **getMetadata1** – gets all Metadata1 values
 - xxxvii. **getMetadata2** – gets all Metadata2 values
 - xxxviii. **getMetadata3** – gets all Metadata3 values

Operations definition

- **insertProduct**

	Attribute	Type	Req	Default	Notes
Request: insert element	productName	String	Y		Must be unique
	checkFrequencyType	DAYS HOURS	N	DAYS	HOURS and available with Premium accounts
	checkFrequencyInterval	Int	N	1	If checkFrequencyType = HOURS, checkFrequencyInterval has to be > 2
	Active	boolean	N	True	If False product's price will not be checked
	Sku	String	N		SKU from your application
	internalId	String	N		ID from your application (should be used as primary key from your DB)
	targetPrice	decimal	N		To be used if Price2Spy should monitor MAP (minimal advertised price) violations
	purchasePrice	decimal	N		To be used if Price2Spy should monitor purchase price
	categoryName	String	N		Product category from your DB)
	brandName	string	N		Product brand from your DB)
	supplierName	string	N		Product supplier from your DB)
	metadata1Name	string	N		Product metadata 1 from your DB)

	metadata2Name	string	N		Product metadata 2 from your DB)
	metadata3Name	string	N		Product metadata 3 from your DB)
	customField1	string	N		Product custom field 1 from your DB)
	customField2	string	N		Product custom field 2 from your DB)
	customField3	string	N		Product custom field 3 from your DB)
	customField4	string	N		Product custom field 4 from your DB)
	customField5	string	N		Product custom field 5 from your DB)
	customField6	string	N		Product custom field 6 from your DB)
	customField7	string	N		Product custom field 7 from your DB)
	customField8	string	N		Product custom field 8 from your DB)
	customField9	string	N		Product custom field 9 from your DB)
	customField10	string	N		Product custom field 10 from your DB)
	comment1	string	N		Product comment 1 from your DB)
	comment2	string	N		Product comment 2 from your DB)
Response	insertProductResponse element	int	Y		If insert was successful, this will contain Price2Spy productID (which needs to be stored in your product DB as well).
Possible Faults	productNameNotUnique				If Product with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Note: Allowed combinations of checkFrequencyType | checkFrequencyInterval are

- DAYS | 1, 2, 3, 5, 7, 14, 30
- HOURS | 3, 4, 6, 8, 12

1. Example SOAP request / response (product successfully inserted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertProductRequest> <mod:insert productName="My 1st product" checkFrequencyType="DAYS" checkFrequencyInterval="1" active="True" sku="SKU-001" internalId="53867" targetPrice="" purchasePrice="" categoryName="Cameras" brandName="Nikon" supplierName=""/> <mod:apiKey>3843737adf8272</mod:apiKey> </mod:insertProductRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <insertProductResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search">130661</insertProductResponse> </soap:Body></pre>

	</soap:Envelope>
--	------------------

- **insertUrl**

	Attribute	Type	Req.	Default	Notes
Request insert element	productId	int	Y		ProductID from Price2Spy (as returned by insertProduct())
	url	string	Y		URL of the product you want to monitor
	Active	boolean	N	True	If False url's price will not be checked
	showInReports	boolean	N	True	If False url will not be shown in reports
	note1	string	N		Note regarding this URL
	note2	string	N		Note regarding this URL
	note3	string	N		Note regarding this URL
Response	insertURLResponse element	int	Y		If insert was successful, this will contain Price2Spy urlID (which needs to be stored in your DB as well).
Possible Faults	maxUrlsForPriceplanExceeded				Exceeded max URLs on your Price2Spy account
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Note: if you're inserting URL for a site which has not been handled in Price2Spy before, a generalError ('site not supported', or 'site already suggested') will be raised. It will take Price2Spy team approx. 1 working day to handle this site. After that insertUrl can be reattempted, and it should succeed.

The above procedure is necessary only when adding the very first URL from a new website.

If insertUrl keeps failing, please contact support@price2spy.com

1. Example SOAP request / response (URL successfully inserted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertUrlRequest> <mod:insert productId="130661" url="http://www.amazon.com/1234" active="true" showInReports="true"/> <mod:apiKey>ddd759c9054a80a</mod:apiKey> </mod:insertUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <insertUrlResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search">241251</insertUrlResponse> </soap:Body> </soap:Envelope></pre>

- Example SOAP request / response (trying to add URL form a new site (www.amazon2.com) which needs to be covered by Price2Spy technical team prior to URL insertion)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertUrlRequest> <mod:insert productId="130661" url="http://www.amazon2.com/1234" active="true" showInReports="true"/> <mod:apiKey>ddd759c9054a80a</mod:apiKey> </mod:insertUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <soap:Fault> <faultcode>soap:Server</faultcode> <faultstring>DataValidationError: We do not support this webshop at the moment, but it has been automatically suggested and we will inform you on results in next 1-2 working days.</faultstring> </soap:Fault> </soap:Body> </soap:Envelope></pre>

- insertCategory (works the same way for insertBrand / insertSupplier)**

	Attribute	Type	Req	Default	Notes
Request: insert element	Name	String	Y		Must be unique
	Active	boolean	Y		If False Category won't be offered in Price2Spy filters
	Priority	Int	Y		Defines the display order of Categories within Price2Spy. Please fill in 0 if all Categories will have same priority (display order will be alphabetical in that case)
	emailRecipient	String	N		List of price change alert recipients for given Category (if more recipients, use ; as separator)
Response	insertCategoryResponse element	int	Y		If insert was successful, this will contain Price2Spy categoryID (which needs to be stored in your product DB as well).
Possible Faults	categoryNameNotUnique				If Category with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

- Example SOAP request / response (Category successfully inserted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertCategoryRequest></pre>
----------------	--

	<pre> <mod:insert name="MyCategory" active="true" priority="0" emailRecipient="xyz1@email.com;xyz2@email.com"/> <mod:apiKey>12345678abc</mod:apiKey> </mod:insertCategoryRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertCategoryResponse>5189</mod:insertCategoryResponse> </soapenv:Body> </soapenv:Envelope> </pre>

- **insertMetadata1 (works the same way for insertBrand / insertSupplier)**

	Attribute	Type	Req	Default	Notes
Request: insert element	Name	String	Y		Must be unique
	Active	boolean	Y		If False Metadata1 won't be offered in Price2Spy filters
	Priority	Int	Y		Defines the display order of Metadata1 within Price2Spy. Please fill in 0 if all Metadata1 will have same priority (display order will be alphabetical in that case)
	emailRecipient	String	N		List of price change alert recipients for given Metadata1 (if more recipients, use ; as separator)
Response	insertMetadata1Response element	int	Y		If insert was successful, this will contain Price2Spy metadata1ID (which needs to be stored in your product DB as well).
Possible Faults	metadata1NameNotUnique				If Metadata1 with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

1. Example SOAP request / response (Metadata1 successfully inserted)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata1Request> <mod:insert name="MyMetadata1" active="true" priority="0" emailRecipient="xyz1@email.com;xyz2@email.com"/> <mod:apiKey>12345678abc</mod:apiKey> </mod:insertMetadata1Request> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata1Response>4321</mod:insertMetadata1Response> </soapenv:Body> </soapenv:Envelope> </pre>

- **insertMetadata2 (works the same way for insertBrand / insertSupplier)**

	Attribute	Type	Req	Default	Notes
Request: insert element	Name	String	Y		Must be unique
	Active	boolean	Y		If False Metadata2 won't be offered in Price2Spy filters
	Priority	Int	Y		Defines the display order of Metadata2 within Price2Spy. Please fill in 0 if all Metadata2 will have same priority (display order will be alphabetical in that case)
	emailRecipient	String	N		List of price change alert recipients for given Metadata2 (if more recipients, use ; as separator)
Response	insertMetadata1Response element	int	Y		If insert was successful, this will contain Price2Spy metadata2ID (which needs to be stored in your product DB as well).
Possible Faults	Metadata2NameNotUnique				If Metadata2 with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

1. Example SOAP request / response (Metadata2 successfully inserted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata2Request> <mod:insert name="MyMetadata2" active="true" priority="0" emailRecipient="xyz1@email.com;xyz2@email.com"/> <mod:apiKey>12345678abc</mod:apiKey> </mod:insertMetadata2Request> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata2Response>4321</mod:insertMetadata2Response> </soapenv:Body> </soapenv:Envelope></pre>

- **insertMetadata3 (works the same way for insertBrand / insertSupplier)**

	Attribute	Type	Req	Default	Notes
Request: insert element	Name	String	Y		Must be unique
	Active	boolean	Y		If False Metadata3 won't be offered in Price2Spy filters
	Priority	Int	Y		Defines the display order of Metadata3 within Price2Spy. Please fill in 0 if all Metadata3 will have same priority (display order

					will be alphabetical in that case)
	emailRecipient	String	N		List of price change alert recipients for given Metadata3 (if more recipients, use ; as separator)
Response	insertMetadata1Response element	int	Y		If insert was successful, this will contain Price2Spy metadata3ID (which needs to be stored in your product DB as well).
Possible Faults	Metadata2NameNotUnique				If Metadata3 with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

1. Example SOAP request / response (Metadata3 successfully inserted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata3Request> <mod:insert name="MyMetadata3" active="true" priority="0" emailRecipient="xyz1@email.com;xyz2@email.com"/> <mod:apiKey>12345678abc</mod:apiKey> </mod:insertMetadata3Request> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:insertMetadata3Response>4321</mod:insertMetadata3Response> </soapenv:Body> </soapenv:Envelope></pre>

- **updateProduct**

	Attribute	Type	Req	Default	Notes
Request update element	productId	int	Y		This must match productID in Price2Spy (returned after insertProduct())
	productName	string	Y		Must be unique
	checkFrequencyType	DAYS HOURS	N	DAYS	HOURS available with Premium accounts
	checkFrequencyInterval	int	N	1	If checkFrequencyType = HOURS, checkFrequencyInterval has to be > 2
	active	boolean	N	True	If False product's price will not be checked
	sku	string	N		SKU from your application
	internalId	string	N		ID from your application (should be used as primary key from your DB)
	targetPrice	decimal	N		To be used if Price2Spy should monitor MAP violations
	purchasePrice	decimal	N		To be used if Price2Spy should monitor purchase price

	categoryName	string	N		Product category from your DB)
	brandName	string	N		Product brand from your DB)
	supplierName	string	N		Product supplier from your DB)
	metadata1Name	string	N		Product metadata 1 from your DB)
	metadata2Name	string	N		Product metadata 2 from your DB)
	metadata3Name	string	N		Product metadata 3 from your DB)
	customField1	string	N		Product custom field 1 from your DB)
	customField2	string	N		Product custom field 2 from your DB)
	customField3	string	N		Product custom field 3 from your DB)
	customField4	string	N		Product custom field 4 from your DB)
	customField5	string	N		Product custom field 5 from your DB)
	customField6	string	N		Product custom field 6 from your DB)
	customField7	string	N		Product custom field 7 from your DB)
	customField8	string	N		Product custom field 8 from your DB)
	customField9	string	N		Product custom field 9 from your DB)
	customField10	string	N		Product custom field 10 from your DB)
	comment1	string	N		Product comment 1 from your DB)
	comment2	string	N		Product comment 2 from your DB)
Response	updateProductResponse				(void) – the presence of this element indicates that updateProduct() was successfully carried out (otherwise a fault would be raised)
Possible Faults	invalidId				If productId supplied was not found in Price2Spy
	productNameNotUnique				If a different Product with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Important: if an attribute is missing from *updateProductRequest* operation, *this attribute's value in Price2Spy account will remain unchanged*

1. Example SOAP request / response (successful product update, with just 1 product attribute set to new value)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateProductRequest></pre>
----------------	---

Only the following attribute will be set to new value	<pre><mod:update productId="130661" productName="Corrected product name" /> <mod:apiKey>ddd756ab54e089f</mod:apiKey> </mod:updateProductRequest> </soapenv:Body> </soapenv:Envelope></pre>
<ul style="list-style-type: none"> productName 	
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

2. Example SOAP request / response (successful product update, with 5 attributes set to empty values)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateProductRequest> <mod:update productId="130661" productName="Corrected product name" checkFrequencyType="DAYS" checkFrequencyInterval="2" active="true" sku="" internalId="" targetPrice="0" purchasePrice="0" categoryName="Cameras" brandName="Nikon" supplierName="" /> <mod:apiKey>ddd756ab54e089f</mod:apiKey> </mod:updateProductRequest> </soapenv:Body> </soapenv:Envelope></pre>
<p>The following attributes will be set to empty values</p> <ul style="list-style-type: none"> sku internalId targetPrice purchasePrice supplierName 	
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

3. Example SOAP request / response (successful product update – targetPrice and purchasePrice have been set to 0)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateProductRequest> <mod:update productId="130661" targetPrice="0.00" purchasePrice="0.00" categoryName="Cameras" brandName="Nikon" supplierName="" /> <mod:apiKey>ddd756ab54e089f</mod:apiKey> </mod:updateProductRequest> </soapenv:Body> </soapenv:Envelope></pre>
<p>The following attributes will be set to 0</p> <ul style="list-style-type: none"> purchasePrice targetPrice 	
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

- updateUrl**

	Attribute	Type	Req.	Default	Notes
Request update element	urlId	int	Y		urlID from Price2Spy (as returned by insertUrl())
	url	string	Y		URL of the product you want to monitor
	counterFailed	int	N	0	Represents successive failed attempts to capture this URL's price
	active	boolean	N	True	If False url's price will not be checked

	showInReports	boolean	N	True	If False url will not be shown in reports
	note1	string	N		Note regarding this URL
	note2	string	N		Note regarding this URL
	note3	string	N		Note regarding this URL
Response	insertURLResponse	int	Y		If insert was successful, this will contain Price2Spy urlID (which needs to be stored in your DB as well).
Possible Faults	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Note: when updating URL, you will not be allowed to change its site. If you need to achieve this, you need to do deleteUrl() (the old one), followed by insertUrl() (new one)

Important: if an attribute is missing from *updateUrlRequest* operation, this attribute's value in Price2Spy account will remain unchanged

1. Example SOAP request / response (URL successfully updated)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateUrlRequest> <mod:update urlId="241251" url="http://www.amazon.com" counterFailed="0" active="true" showInReports="true"/> <mod:apiKey>ddd759c9054a8</mod:apiKey> </mod:updateUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

2. Example SOAP request / response (URL was not updated – tried to change the site)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateUrlRequest> <mod:update urlId="241251" url="http://www.buy.com" counterFailed="0" active="true" showInReports="true"/> <mod:apiKey>ddd759c9054a8</mod:apiKey> </mod:updateUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <soap:Fault> <faultcode>soap:Server</faultcode> <faultstring>DataValidationError: You can change URL just for this Webshop. If you want to change the Webshop, delete existing URL and add a new one.</faultstring> </soap:Fault> </soap:Body> </soap:Envelope></pre>

3. Example SOAP request / response (URL successfully updated, counterFailed value left unchanged)

<p>Request</p> <p>Please note that we have omitted counterFailed attribute – meaning that it's value in Price2Spy will remain unchanged</p>	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateUrlRequest> <mod:update urlId="241251" url="http://www.amazon.com" active="true" showInReports="true"/> <mod:apiKey>ddd759c9054a8</mod:apiKey> </mod:updateUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
<p>Response</p>	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

- **updateCategory (works the same way for updateBrand / updateSupplier / updateMetadata1 / updateMetadata2 / updateMetadata3)**

	Attribute	Type	Req	Default	Notes
Request: update element	categoryId	Int	Y		ID of the category to be updated
	Name	String	Y		Must be unique
	Active	boolean	Y		If False Category won't be offered in Price2Spy filters
	Priority	Int	Y		Defines the display order of Categories within Price2Spy. Please fill in 0 if all Categories will have same priority (display order will be alphabetical in that case)
	emailRecipient	String	N		List of price change alert recipients for given Category (if more recipients, use ; as separator)
Response	updateCategoryResponse element				If insert was successful, the response element will be empty – otherwise a Fault will be thrown
Possible Faults	categoryNameNotUnique				If Category with same name already exists in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

1. Example SOAP request / response (Category successfully updated)

<p>Request</p> <p>Only the following attribute will be set to new value</p> <ul style="list-style-type: none"> • Name 	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:updateCategoryRequest> <mod:update categoryId="5789" name="My new Category Name"/> <mod:apiKey>1234</mod:apiKey> </mod:updateCategoryRequest> </soapenv:Body> </soapenv:Envelope></pre>
---	---

Response	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body/> </soapenv:Envelope></pre>
-----------------	---

- **deleteProduct**

	Element	Type	Req	Default	Notes
Request deleteProductRequest element	productId	int	Y		This must match productID in Price2Spy (returned after insertProduct())
Response	deleteProductResponse				(void) – the presence of this element indicates that deleteProduct() was successfully carried out (otherwise a fault would be raised)
Possible Faults	invalidId				If productId supplied was not found in Price2Spy
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Note: deleteProduct() will also delete all underlying URLs and their price history

Example SOAP request / response (product successfully deleted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:deleteProductRequest> <mod:productId>130661</mod:productId> <mod:apiKey>ddd759c9054a80a45f20e196ab54e089f</mod:apiKey> </mod:deleteProductRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

- **deleteUrl**

	Element	Type	Req	Default	Notes
Request	urlId	int	Y		This must match urlID in Price2Spy (returned after insertProduct())
Response	deleteUrlResponse				(void) – the presence of this element indicates that deleteUrl() was successfully carried out (otherwise a fault would be raised)
Possible Faults	invalidId				If urlId supplied was not found in Price2Spy

	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

Note: deleteUrl() will also delete it's price history

Example SOAP request / response (product successfully deleted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:deleteUrlRequest> <mod:urlId>241251</mod:urlId> <mod:apiKey>474574357</mod:apiKey> </mod:deleteUrlRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>

- **deleteCategory (works the same way for deleteBrand / deleteSupplier / deleteMetadata1 / deleteMetadata2 / deleteMetadata3)**

	Element	Type	Req	Default	Notes
Request deleteCategoryRequest element	categoryId	int	Y		This must match categoryID in Price2Spy (returned after insertCategory())
Response	deleteCategoryResponse				(void) – the presence of this element indicates that deleteCategory() was successfully carried out (otherwise a fault would be raised)
Possible Faults	invalidId				If categoryId supplied was not found in Price2Spy
	categoryNotDeletable				A category can be deleted only if it has no products associated with it
	dataValidationError				An error detected with your data. Refer to error details
	generalError				General error detected. Refer to error details

1. Example SOAP request / response (product successfully deleted)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:deleteCategoryRequest> <mod:categoryId>5891</mod:categoryId> <mod:productId> <mod:apiKey>ddd759c9054a80a45f20e196ab54e089f</mod:apiKey> </mod:productId> </mod:deleteCategoryRequest> </soapenv:Body> </soapenv:Envelope></pre>
----------------	---

Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body/> </soap:Envelope></pre>
-----------------	---

- **getProducts**

	attributes	Type	Req	Default	Notes
Request	searchProducts	complex	Y		See below for searchProductType parameters
Response	List of product elements	complex			List of productType elements returned by search
Possible Faults	generalError				General error detected. Refer to error details

SearchProducts contains one or more of the following elements (if more elements are present, they will form logical 'AND' condition). Therefore, if you need to specify just one attribute in the search, you need to omit all other attributes

- searchMode – LIKE | EQUAL (if omitted, the default is LIKE)
 - LIKE will return all products which contain searched value as a substring (in SQL terminology, this would be represented by operator LIKE)
 - EQUAL will return only products with exact attribute value that has been searched (in SQL terminology, this would be represented by operator =)
- productId - int
- productName – string
- url – string
- sku – string
- internalId – string
- categoryName – string
- categoryId – int
- brandName – string
- brandId – int
- supplierName – string
- supplierId – int
- Metadata1Id – int
- Metadata1Name – string
- Metadata2Id – int
- Metadata2Name – string
- Metadata3Id – int
- Metadata3Name – string
- dateChangeFrom – datetime (refers to timestamp of the last price / availability change detected for this product)
- dateChangeTo – datetime (refers to timestamp of the last price / availability change detected for this product)

Note: If you are using search with date attributes (dateChangeFrom, dateChangeTo), search result will contain only measurement records in given timeframe.

ProductType complex element contains the following elements

- productId - int (Product's ID in Price2Spy)
- productName – string
- checkFrequencyType = DAYS | HOURS
- checkFrequencyInterval - int
- active – boolean
- showInReports – boolean
- sku – string
- internalId – string
- categoryName – string
- categoryId – int (categoryID in Price2Spy, if defined)
- brandName – string
- brandId – int (brandID in Price2Spy, if defined)
- supplierName – string
- supplierId – int (supplierID in Price2Spy, if defined)
- Metadata1Name – string
- Metadata1Id – int (Metadata1Id in Price2Spy, if defined)
- Metadata2Name – string
- Metadata2Id – int (Metadata2Id in Price2Spy, if defined)
- Metadata3Name – string
- Metadata3Id – int (Metadata3Id in Price2Spy, if defined)
- minPrice – price (complex type) – minimal price of given product on all URLs
- maxPrice – price (complex type) – maximal price of given product on all URLs
- lastChecked – datetime (timestamp of last price check performed by Price2Spy)
- nextChecked – datetime (timestamp of next price check to be performed by Price2Spy)
- lastChange – datetime (timestamp of last price change detected by Price2Spy)
- targetPrice – decimal (usually represents MAP price for this product)
- purchasePrice – decimal (usually represents purchase price for this product)
- dateAdded – datetime (timestamp when the product was added to Price2Spy)
- dateModified – datetime (timestamp when the product was modified in Price2Spy)
- addedById – int (ID of user that added the product into Price2Spy)
- modifiedById – int (ID of user that modified the product in Price2Spy)

PriceType complex element contains the following elements

- amount – decimal (price detected by Price2Spy)
- currency – string (3-letter ISO code for currency on given site)

1. Example SOAP request / response (search all products, list of all products on your account returned)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getProductsRequest> <sear:search/> <sear:apiKey>ddd75f</sear:apiKey> </sear:getProductsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getProductsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:product> <ns3:productId>1473</ns3:productId> <ns3:productName>Canon 600D body only</ns3:productName> <ns3:checkFrequencyType>DAYS</ns3:checkFrequencyType> <ns3:checkFrequencyInterval>1</ns3:checkFrequencyInterval> <ns3:active>true</ns3:active> <ns3:showInReports>true</ns3:showInReports> <ns3:minPrice> <ns3:amount>425.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:minPrice> <ns3:maxPrice> <ns3:amount>499.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:maxPrice> <ns3:lastChecked>2012-09-20T10:16:49.217+02:00</ns3:lastChecked> <ns3:nextChecked>2012-09-21T10:10:00+02:00</ns3:nextChecked> <ns3:lastChange>2012-09-20T10:16:49.220+02:00</ns3:lastChange> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:20:05+01:00</ns3:dateModified> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:product> <ns3:product> </ns3:product> </ns3:getProductsResponse> </soap:Body> </soap:Envelope> </pre>

2. Example SOAP request / response (search for a single product, based on its ID)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getProductsRequest> <sear:search productId="75296"/> <sear:apiKey>ddd759c9054a80a45f089f</sear:apiKey> </sear:getProductsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getProductsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:product> <ns3:productId>75296</ns3:productId> </pre>

	<pre> <ns3:productName>Nikon D600 Digital Camera with 24-85mm f/3.5-4.5G ED VR Lens</ns3:productName> <ns3:checkFrequencyType>DAYS</ns3:checkFrequencyType> <ns3:checkFrequencyInterval>1</ns3:checkFrequencyInterval> <ns3:active>>true</ns3:active> <ns3:showInReports>true</ns3:showInReports> <ns3:sku/> <ns3:internalId/> <ns3:minPrice> <ns3:amount>2484.24</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:minPrice> <ns3:maxPrice> <ns3:amount>2868.84</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:maxPrice> <ns3:lastChecked>2012-09-20T04:50:06.400+02:00</ns3:lastChecked> <ns3:nextChecked>2012-09-20T21:04:27.727+02:00</ns3:nextChecked> <ns3:lastChange>2012-09-20T04:50:05.617+02:00</ns3:lastChange> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:20:05+01:00</ns3:dateModified> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:product> </ns3:getProductsResponse> </soap:Body> </soap:Envelope> </pre>

3. Example SOAP request / response (search for all products from given category)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getProductsRequest> <sear:search categoryName="Printer"/> <sear:apiKey>ddd759c9054a80a45f20e1f786c8b96ab54e089f</sear:apiKey> </sear:getProductsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getProductsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:product> <ns3:productId>16678</ns3:productId> <ns3:productName>HP Envy 114 E-All-in-one Wireless Thermal Printer</ns3:productName> <ns3:checkFrequencyType>DAYS</ns3:checkFrequencyType> <ns3:checkFrequencyInterval>1</ns3:checkFrequencyInterval> <ns3:active>true</ns3:active> <ns3:showInReports>true</ns3:showInReports> <ns3:sku/> <ns3:internalId/> <ns3:categoryName>Printer</ns3:categoryName> <ns3:categoryId>439</ns3:categoryId> <ns3:brandName>HP</ns3:brandName> <ns3:brandId>1085</ns3:brandId> <ns3:minPrice> <ns3:amount>220.58</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:minPrice> <ns3:maxPrice> <ns3:amount>272.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:maxPrice> <ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked> <ns3:nextChecked>2012-09-21T08:18:22.997+02:00</ns3:nextChecked> </pre>

	<pre><ns3:lastChange>2012-09-20T08:21:08.997+02:00</ns3:lastChange> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:20:05+01:00</ns3:dateModified> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:product> </ns3:getProductsResponse> </soap:Body> </soap:Envelope></pre>
--	---

- **getUrls**

	Attributes	Type	Req	Default	Notes
Request search element	getUrlsRequest	complex	Y		See below for search parameters
Response	Urls	complex			List of urlType elements returned by search
Possible Faults	generalError				General error detected. Refer to error details

Search contains the following element

- searchMode – LIKE | EQUAL (if omitted, the default is LIKE)
 - LIKE will return all products which contain searched value as a substring (in SQL terminology, this would be represented by operator LIKE)
 - EQUAL will return only products with exact attribute value searched for (in SQL terminology, this would be represented by operator =)
- id - int
- idMode – string (ID | SKU | INTERNAL_ID | NAME | ANY– specifies which product field will be used for search)
 - ID = product with this Price2Spy’s ID will be returned
 - SKU = products with matching SKU will be returned
 - INTERNAL_ID = products with matching INTERNAL_ID will be returned
 - NAME = products with matching NAME will be returned
 - ANY = products where either SKU, INTERNAL_ID or NAME match will be returned

UrlType complex element contains the following elements

- urlId - int (URL’s ID in Price2Spy)
- url – string
- productName – string (product name from the measured URL, as found on the website)
- siteHumanName – string – site’s name
- siteCountryISOCode – string – ISO code of site’s country
- showInReports – Boolean
- active – Boolean
- counterFailed – int
- lastError - string (if present, shows the last error Price2Spy report for a failed price check)
- lastChecked – datetime (timestamp of the last price checked performed by Price2Spy)
- lastMeasurement – measurement (complex type representing last successful price check)
- secondToLastMeasurement – measurement (complex type representing penultimate successful price check)

- image1Url – string (URL of the 1st image of the URL)
- image2Url – string (URL of the 2nd image of the URL)
- image3Url – string (URL of the 3rd image of the URL)
- note1 – string (1st URL note)
- note2 – string (2nd URL note)
- note3 – string (3rd URL note)
- dateAdded – datetime (timestamp when the URL was added to Price2Spy)
- dateModified – datetime (timestamp when the URL was modified in Price2Spy)
- addedById – int (ID of user that added the URL into Price2Spy)
- modifiedById – int (ID of user that modified the URL in Price2Spy)

MeasurementType complex element contains the following elements

- dateChecked – datetime (timestamp of the price check)
- price – price (complex type PriceType)
- available – boolean (was the product available on the site)
- shippingCost – decimal (returned only if monitored on user’s account)
- shippingTime – string (returned only if monitored on user’s account)
- shippingDetails – string (returned only if monitored on user’s account)
- stockStatus – string (returned only if monitored on user’s account)
- seller – string (returned only if monitored on user’s account)
- shelfPrice – price (returned only if monitored on user’s account)
- isOnPromotion – boolean (returned only if monitored on user’s account)
- promotionText – string (returned only if monitored on user’s account)
- screenshotUrl – string (if screenshot was taken)
- customField1 – string (returned only if monitored on user’s account)
- customField2 – string (returned only if monitored on user’s account)
- customField3 – string (returned only if monitored on user’s account)
- customField4 – string (returned only if monitored on user’s account)
- customField5 – string (returned only if monitored on user’s account)
- customField6 – string (returned only if monitored on user’s account)
- customField7 – string (returned only if monitored on user’s account)
- customField8 – string (returned only if monitored on user’s account)
- customField9 – string (returned only if monitored on user’s account)
- customField10 – string (returned only if monitored on user’s account)

1. Example SOAP request / response (search for all URLs for given product)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getUrlsRequest> <sear:search id="16678" idMode="ID"/> <sear:apiKey>f20e1f786c8b96ab54e089f</sear:apiKey> </sear:getUrlsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
----------------	---

Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getUrlsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:url> <ns3:urlId>45836</ns3:urlId> <ns3:url>http://www.amazon.com/HP-Envy-E-all---one-Printer/dp/B005PYCLXA/ref=sr_1_1?ie=UTF8&camp;qid=1336458175&camp;sr=8-1</ns3:url> <ns3:productName>HP Envy 114 E-all-in-one Printer - D411C (Black) </ns3:productName> <ns3:siteHumanName>Amazon.com</ns3:siteHumanName> <ns3:siteCountryISOCODE>US</ns3:siteHumanName> <ns3:showInReports>true</ns3:showInReports> <ns3:active>true</ns3:active> <ns3:counterFailed>0</ns3:counterFailed> <ns3:lastError/> <ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked> <ns3:lastMeasurement> <ns3:dateChecked>2012-09-20T08:21:08.997+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>220.58</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> <ns3:screenshotUrl>https://screenshots.price2spy.com/379/53982227-a5g8-420c-a12d-e415333a0ed8.png</ns3:screenshotUrl> <ns3:shippingCost>3.50</ns3:shippingCost> <ns3:shippingTime>3-4 working days</ns3:shippingTime> </ns3:lastMeasurement> <ns3:image1Url> https://www.amazon.com/images/I/61K%2B0mlsGXL_AC_SL1300_.jpg </ns3:image1Url> <ns3:note1>This is a note regarding the above URL from Amazon</ns3:note1> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> <ns3:url> <ns3:urlId>45835</ns3:urlId> <ns3:url>http://www.xyz.com.au/products/HP-Envy-114-E-%252dAll%252din%252done-Wireless-Thermal-Printer.html</ns3:url> <ns3:productName>HP-Envy-114-E-All-in-one-Wireless-Thermal-Printer</ns3:productName> <ns3:siteHumanName>Discount Digital Photographics</ns3:siteHumanName> <ns3:showInReports>true</ns3:showInReports> <ns3:active>true</ns3:active> <ns3:counterFailed>9</ns3:counterFailed> <ns3:lastError>Products price format is changed.</ns3:lastError> <ns3:lastChecked>2012-09-20T08:21:08.990+02:00</ns3:lastChecked> <ns3:lastMeasurement> <ns3:dateChecked>2012-06-22T08:21:13.780+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>272.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> </ns3:lastMeasurement> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> </ns3:getUrlsResponse> </soap:Body> </soap:Envelope> </pre>

- **getMeasurements**

	Attributes	Type	Req	Default	Notes
Request search element	urlId	int	Y		urlId from Price2Spy
	dateCheckedFrom	datetime	N		Get measurements which occurred only after this date
	dateCheckedTo	datetime	N		Get measurements which occurred only before this date
Response	List of measurement elements	complex			List of measurementType elements
Possible Faults	generalError				General error detected. Refer to error details

This operation returns all price checks (measurements) for given urlId.
See above for **measurementType** definition

1. Example SOAP request / response (search for all price checks for given URL)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getMeasurementsRequest> <sear:search urlId="45836"/> <sear:apiKey>ddd759c9054a80a45fb54e089f</sear:apiKey> </sear:getMeasurementsRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getMeasurementsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:measurement> <ns3:dateChecked>2012-09-20T08:21:08.997+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>220.58</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> </ns3:measurement> <ns3:measurement> <ns3:dateChecked>2012-09-19T18:20:00.390+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>208.53</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> <ns3:screenshotUrl>https://screenshots.price2spy.com/379/53982227-a5g8-420c-a12d- e415333a0ed8.png</ns3:screenshotUrl> </ns3:measurement> </ns3:getMeasurementsResponse> </soap:Body> </soap:Envelope></pre>

- **getCurrentPricingData**

	Element	Type	Req	Default	Notes
Request	Seach	complex	Y		See above for seachProductType parameters
Response	productswithUrls	complex			List of productWithUrlsType elements returned by search
Possible Faults	generalError				General error detected. Refer to error details

Note: If you are using search with date attributes (dateChangeFrom, dateChangeTo), search result will contain only measurement records in given timeframe.

ProductWithUrlsType contains list of **Product** elements, and subelements **url** (list of products matching the search, including all its URLs (URLs containing last and penultimate price checks), and approved Automatch URLs.

1. Example SOAP request / response (get current pricing for all products matching search criteria (there has to be at least one search criterion supplied))

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getCurrentPricingDataRequest> <sear:search categoryId="439" /> <sear:apiKey>34343 </sear:apiKey> </sear:getCurrentPricingDataRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getCurrentPricingDataResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:products> <ns3:product> <ns3:productId>16678</ns3:productId> <ns3:productName>HP Envy 114 E-All-in-one Wireless Thermal Printer</ns3:productName> <ns3:checkFrequencyType>DAYS</ns3:checkFrequencyType> <ns3:checkFrequencyInterval>1</ns3:checkFrequencyInterval> <ns3:active>true</ns3:active> <ns3:showInReports>true</ns3:showInReports> <ns3:sku/> <ns3:internalId/> <ns3:categoryName>Printer</ns3:categoryName> <ns3:categoryId>439</ns3:categoryId> <ns3:brandName>HP</ns3:brandName> <ns3:brandId>1085</ns3:brandId> <ns3:metadata1Name>Metadata1Name</ns3:metadata1Name> <ns3:metadata1Id>4396</ns3:metadata1Id> <ns3:metadata2Name>Metadata2Name</ns3:metadata2Name> <ns3:metadata2Id>4397</ns3:metadata2Id> <ns3:metadata3Name>Metadata3Name</ns3:metadata3Name> <ns3:metadata3Id>4398</ns3:metadata3Id> <ns3:minPrice> <ns3:amount>220.58</ns3:amount></pre>

```
<ns3:currency>AUD</ns3:currency>
</ns3:minPrice>
<ns3:maxPrice>
  <ns3:amount>272.00</ns3:amount>
  <ns3:currency>AUD</ns3:currency>
</ns3:maxPrice>
<ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked>
<ns3:nextChecked>2012-09-21T08:18:22.997+02:00</ns3:nextChecked>
<ns3:lastChange>2012-09-20T08:21:08.997+02:00</ns3:lastChange>
<ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded>
<ns3:dateModified>2019-02-14T10:20:05+01:00</ns3:dateModified>
<ns3:addedByld>1234</ns3:addedByld>
<ns3:modifiedByld>1234</ns3:modifiedByld>
<ns3:urls>
  <ns3:url>
    <ns3:urlId>45836</ns3:urlId>
    <ns3:url>http://www.amazon.com/HP-Envy-E-all---one-Printer/dp/B005PYCLXA/ref=sr\_1\_1?ie=UTF8&qid=1336458175&sr=8-1</ns3:url>
  <ns3:productName>HP Envy 114 E-all-in-one Printer - D411C (Black)
</ns3:productName>
  <ns3:siteHumanName>Amazon.com</ns3:siteHumanName>
  <ns3:siteCountryISOCode>US</ns3:siteHumanName>
  <ns3:showInReports>true</ns3:showInReports>
  <ns3:active>true</ns3:active>
  <ns3:counterFailed>0</ns3:counterFailed>
  <ns3:lastError/>
  <ns3:screenshotUrl>https://screenshots.price2spy.com/379/53982227-a5g8-420c-a12d-e415333a0ed8.png</ns3:screenshotUrl>
  <ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked>
  <ns3:lastMeasurement>
    <ns3:dateChecked>2012-09-20T08:21:08.997+02:00</ns3:dateChecked>
    <ns3:price>
      <ns3:amount>220.58</ns3:amount>
      <ns3:currency>AUD</ns3:currency>
    </ns3:price>
    <ns3:available>true</ns3:available>
  </ns3:lastMeasurement>
  <ns3:automatchUrl>
    <ns3:automatchUrlId>456281</ns3:automatchUrlId>
    <ns3:url>http://www.amazon.com/HP-Envy-E-all---one-Printer/dp/B005PYCLXA/ref=sr\_1\_1?ie=UTF8&qid=1336458175&sr=8-1</ns3:url>
    <ns3:matchAccuracy>92</ns3:matchAccuracy>
    <ns3:dateChecked>2012-09-20T08:21:08.993+02:00</ns3:dateChecked>
    <approved>true</ns3:approved>
    <ns3:dateApproved>2012-09-20T08:21:08.993+02:00</ns3:dateApproved>
  <ns3:automatchUrlFields>
    <ns3:automatchUrlField>
      <ns3:name>price</ns3:name>
      <ns3:value>220.58</ns3:value>
    </ns3:automatchUrlField>
    <ns3:automatchUrlField>
      <ns3:name>name</ns3:name>
      <ns3:value>HP Envy 114 E-All-in-one Wireless Thermal Printer</ns3:value>
    </ns3:automatchUrlField>
  </ns3:automatchUrlFields>
  <ns3:dateAdded>2012-09-20T08:21:08.993+02:00</ns3:dateAdded>
  <ns3:dateModified>2012-09-20T08:21:08.993+02:00</ns3:dateModified>
  <ns3:addedByld>1234</ns3:addedByld>
  <ns3:modifiedByld>1234</ns3:modifiedByld>
</ns3:automatchUrl>
```

	<pre> <ns3:image1Url> https://www.amazon.com/images/I/61K%2B0mlsGXL_AC_SL1300_.jpg </ns3:image1Url> <ns3:note1>This is a note regarding the above URL from Amazon</ns3:note1> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> <ns3:url> <ns3:urllId>45835</ns3:urllId> <ns3:url>http://www.xyz.com.au/products/HP-Envy-114- E%252dAll%252din%252done-Wireless-Thermal-Printer.html</ns3:url> <ns3:productName>HP-Envy-114-E-All-in-one-Wireless-Thermal- Printer</ns3:productName> <ns3:siteHumanName>Discount Digital Photographics</ns3:siteHumanName> <ns3:siteCountryISOCode>AU</ns3:siteCountryISOCode> <ns3:showInReports>true</ns3:showInReports> <ns3:active>true</ns3:active> <ns3:counterFailed>9</ns3:counterFailed> <ns3:lastError>Products price format is changed.</ns3:lastError> <ns3:lastChecked>2012-09-20T08:21:08.990+02:00</ns3:lastChecked> <ns3:lastMeasurement> <ns3:dateChecked>2012-06-22T08:21:13.780+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>272.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> </ns3:lastMeasurement> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> </ns3:urls> </ns3:product> </ns3:products> </ns3:getCurrentPricingDataResponse> </soap:Body> </soap:Envelope> </pre>

2. Example SOAP request / response (get current pricing for all products matching search criteria – but this time on a product which contains additional captured fields (like shipping cost, seller etc). Please note that the fields containing no data will not be returned (in other words: if there is no information on shipping time, <shippingTime> element will not be returned)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getCurrentPricingDataRequest> <sear:search categoryId="431" /> <sear:apiKey>34343</sear:apiKey> </sear:getCurrentPricingDataRequest> </soapenv:Body> </pre>
----------------	--

<p>Response</p>	<pre> </soapenv:Envelope> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getCurrentPricingDataResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:products> <ns3:product> <ns3:productId>16678</ns3:productId> <ns3:productName>HP Envy 114 E-All-in-one Wireless Thermal Printer</ns3:productName> <ns3:checkFrequencyType>DAYS</ns3:checkFrequencyType> <ns3:checkFrequencyInterval>1</ns3:checkFrequencyInterval> <ns3:active>true</ns3:active> <ns3:showInReports>true</ns3:showInReports> <ns3:sku/> <ns3:internalId/> <ns3:categoryName>Printer</ns3:categoryName> <ns3:categoryId>439</ns3:categoryId> <ns3:brandName>HP</ns3:brandName> <ns3:brandId>1085</ns3:brandId> <ns3:metadata1Name>Metadata1Name</ns3:metadata1Name> <ns3:metadata1Id>4396</ns3:metadata1Id> <ns3:metadata2Name>Metadata2Name</ns3:metadata2Name> <ns3:metadata2Id>4397</ns3:metadata2Id> <ns3:metadata3Name>Metadata3Name</ns3:metadata3Name> <ns3:metadata3Id>4398</ns3:metadata3Id> <ns3:minPrice> <ns3:amount>220.58</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:minPrice> <ns3:maxPrice> <ns3:amount>272.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:maxPrice> <ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked> <ns3:nextChecked>2012-09-21T08:18:22.997+02:00</ns3:nextChecked> <ns3:lastChange>2012-09-20T08:21:08.997+02:00</ns3:lastChange> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:20:05+01:00</ns3:dateModified> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> <ns3:urls> <ns3:url> <ns3:urlId>45836</ns3:urlId> <ns3:url>http://www.amazon.com/HP-Envy-E-all---one-Printer/dp/B005PYCLXA/ref=sr_1_1?ie=UTF8&amp;qid=1336458175&amp;sr=8-1</ns3:url> <ns3:productName>HP Envy 114 E-all-in-one Printer - D411C (Black) </ns3:productName> <ns3:siteHumanName>Amazon.com</ns3:siteHumanName> <ns3:siteCountryISOCode>US</ns3:siteHumanName> <ns3:showInReports>true</ns3:showInReports> <ns3:active>true</ns3:active> <ns3:counterFailed>0</ns3:counterFailed> <ns3:lastError/> <ns3:screenshotUrl>https://screenshots.price2spy.com/379/53982227-a5g8-420c-a12d-e415333a0ed8.png</ns3:screenshotUrl> <ns3:lastChecked>2012-09-20T08:21:08.993+02:00</ns3:lastChecked> <ns3:lastMeasurement> <ns3:dateChecked>2012-09-20T08:21:08.997+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>220.58</ns3:amount> </ns3:price> </ns3:lastMeasurement> </ns3:urls> </ns3:product> </ns3:products> </ns3:getCurrentPricingDataResponse> </soap:Body> </soap:Envelope> </pre>
------------------------	---

	<pre> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> <ns3:shippingCost>3.50</ns3:shippingCost> <ns3:shippingTime>3-4 working days</ns3:shippingTime> <ns3:stockStatus>50</ns3:stockStatus> <ns3:customField1>-30</ns3:customField1> <ns3:shelfPrice>250.00</ns3: shelfPrice> <ns3:isOnPromotion>True</ns3:isOnPromotion> <ns3:promotionText>Valentine's day discount</ns3: promotionText> </ns3:lastMeasurement> <ns3:automatchUrl> <ns3:automatchUrlId>456281</ns3:automatchUrlId> <ns3:url>http://www.amazon.com/HP-Envy-E-all---one- Printer/dp/B005PYCLXA/ref=sr_1_1?ie=UTF8&qid=1336458175&sr=8-1</ns3:url> <ns3:matchAccuracy>92</ns3:matchAccuracy> <ns3:dateChecked>2012-09-20T08:21:08.993+02:00</ns3:dateChecked> <approved>true</ns3:approved> <ns3:dateApproved>2012-09-20T08:21:08.993+02:00</ns3:dateApproved> <ns3:automatchUrlFields> <ns3:automatchUrlField> <ns3:name>price</ns3:name> <ns3:value>220.58</ns3:value> </ns3:automatchUrlField> <ns3:automatchUrlField> <ns3:name>name</ns3:name> <ns3:value>HP Envy 114 E-All-in-one Wireless Thermal Printer</ns3:value> </ns3:automatchUrlField> </ns3:automatchUrlFields> <ns3:dateAdded>2012-09-20T08:21:08.993+02:00</ns3:dateAdded> <ns3:dateModified>2012-09-20T08:21:08.993+02:00</ns3:dateModified> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:automatchUrl> <ns3:image1Url> https://www.amazon.com/images/I/61K%2B0mlsGXL_AC_SL1300_.jpg </ns3:image1Url> <ns3:note1>This is a note regarding the above URL from Amazon</ns3:note1> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> <ns3:url> <ns3:urlId>45835</ns3:urlId> <ns3:url>http://www.xyz.com.au/products/HP-Envy-114- E%252dAll%252din%252done-Wireless-Thermal-Printer.html</ns3:url> <ns3:productName>HP-Envy-114-E-All-in-one-Wireless-Thermal- Printer</ns3:productName> <ns3:siteHumanName>Discount Digital Photographics</ns3:siteHumanName> <ns3:siteCountryISOCode>AU</ns3:siteHumanName> <ns3:showInReports>true</ns3:showInReports> <ns3:active>true</ns3:active> <ns3:counterFailed>9</ns3:counterFailed> <ns3:lastError>Products price format is changed.</ns3:lastError> <ns3:lastChecked>2012-09-20T08:21:08.990+02:00</ns3:lastChecked> <ns3:lastMeasurement> <ns3:dateChecked>2012-06-22T08:21:13.780+02:00</ns3:dateChecked> <ns3:price> <ns3:amount>272.00</ns3:amount> <ns3:currency>AUD</ns3:currency> </ns3:price> <ns3:available>true</ns3:available> </ns3:lastMeasurement> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> </pre>
--	---

	<pre> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:url> </ns3:urls> </ns3:product> </ns3:products> </ns3:getCurrentPricingDataResponse> </soap:Body> </soap:Envelope> </pre>

- **getBrands**

	Attributes	Type	Req	Default	Notes
Request	getBrandsRequest	complex	Y		See below for parameters
Response	brand	complex			List of BrandType elements
Possible Faults	generalError				General error detected. Refer to error details

BrandType complex element contains the following elements

- id - int (brand's ID in Price2Spy)
- name – String
- active – boolean
- emailRecipient – String (list of emails where price alert notifications are sent, 'if 'Split emails by brand' option is selected in settings)
- dateAdded – datetime (timestamp when the brand was added to Price2Spy)
- dateModified – datetime (timestamp when the brand was modified in Price2Spy)
- addedById – int (ID of user that added the brand into Price2Spy)
- modifiedById – int (ID of user that modified the brand in Price2Spy)

Example SOAP request / response (get all brands)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getBrandsRequest> <sear:apiKey>88a63363e59afa66266ee8aa69a7aacd6b29cab</sear:apiKey> </sear:getBrandsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getBrandsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:brand> <ns3:id>435</ns3:id> <ns3:name>brand name one</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient>one@domain.com; two@domain.com</ns3:emailRecipient> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:brand> <ns3:brand> </pre>

	<pre> <ns3:id>3872</ns3:id> <ns3:name> brand name two</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient/> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:brand> </ns3:getBrandsResponse> </soap:Body> </soap:Envelope> </pre>
--	--

- **getCategories**

	Attributes	Type	Req	Default	Notes
Request	getCategoriesRequest	complex	Y		See below for parameters
Response	brand	complex			List of CategoryType elements
Possible Faults	generalError				General error detected. Refer to error details

CategoryType complex element contains the following elements:

- id - int (category's ID in Price2Spy)
- name – String
- active – boolean
- emailRecipient – String (list of emails where price alert notifications are sent, 'if 'Split emails by category' option is selected in settings)
- dateAdded – datetime (timestamp when the category was added to Price2Spy)
- dateModified – datetime (timestamp when the category was modified in Price2Spy)
- addedById – int (ID of user that added the category into Price2Spy)
- modifiedById – int (ID of user that modified the category in Price2Spy)

Example SOAP request / response (get all categories)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getCategoriesRequest> <sear:apiKey>88a63363e59afa66266ee8aa69a7aacd6b29c29f</sear:apiKey> </sear:getCategoriesRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getCategoriesResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:category> <ns3:id>848</ns3:id> <ns3:name>category one</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient>one@domain.com; two@domain.com</ns3:emailRecipient> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:category> </pre>

	<pre> <ns3:category> <ns3:id>86</ns3:id> <ns3:name> category two</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient/> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:category> </ns3:getCategoriesResponse> </soap:Body> </soap:Envelope> </pre>
--	--

- **getSuppliers**

	Attributes	Type	Req	Default	Notes
Request	getSuppliersRequest	complex	Y		See below for parameters
Response	supplier	complex			List of SupplierType elements
Possible Faults	generalError				General error detected. Refer to error details

SupplierType complex element contains the following elements:

- id - int (supplier's ID in Price2Spy)
- name – String
- active – boolean
- emailRecipient – String (list of emails where price alert notifications are sent, 'if 'Split emails by supplier' option is selected in settings)
- dateAdded – datetime (timestamp when the supplier was added to Price2Spy)
- dateModified – datetime (timestamp when the supplier was modified in Price2Spy)
- addedById – int (ID of user that added the supplier into Price2Spy)
- modifiedById – int (ID of user that modified the supplier in Price2Spy)

Example SOAP request / response (get all suppliers)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getCategoriesRequest> <sear:apiKey>88a63363e59afa66266ee8aa69a7aacd6b29c29f</sear:apiKey> </sear:getCategoriesRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getSuppliersResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:supplier> <ns3:id>39</ns3:id> <ns3:name>supplier one</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient>one@domain.com; two@domain.com</ns3:emailRecipient> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:supplier> </ns3:getSuppliersResponse> </soap:Body> </soap:Envelope> </pre>

	<pre> </ns3:supplier> <ns3:supplier> <ns3:id>4142</ns3:id> <ns3:name>supplier two</ns3:name> <ns3:active>true</ns3:active> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:supplier> </ns3:getSuppliersResponse> </soap:Body> </soap:Envelope> </pre>
--	--

- **getPriceBuckets**

	Attributes	Type	Req	Default	Notes
Request	getSuppliersRequest	complex	Y		See below for parameters
Response	priceBucket	complex			List of PriceBucketType elements
Possible Faults	generalError				General error detected. Refer to error details

PriceBucketType complex element contains the following elements

- id - int (price bucket's ID in Price2Spy)
- name – String
- priceFrom – BigDecimal
- priceTo – BigDecimal
- dateAdded – datetime (timestamp when the price bucket was added to Price2Spy)
- dateModified – datetime (timestamp when the price bucket was modified in Price2Spy)
- addedById – int (ID of user that added the price bucket into Price2Spy)
- modifiedById – int (ID of user that modified the price bucket in Price2Spy)

Example SOAP request / response (get all price buckets)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getPriceBucketsRequest> <sear:apiKey>88a63363e59afa66266ee8aa69a7aacd6b29c29f</sear:apiKey> </sear:getPriceBucketsRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getPriceBucketsResponse xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:priceBucket> <ns3:id>28</ns3:id> <ns3:name>Cheap</ns3:name> <ns3:priceFrom>1.00</ns3:priceFrom> <ns3:priceTo>1000.00</ns3:priceTo> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </pre>

	<pre> </ns3:priceBucket> <ns3:priceBucket> <ns3:id>29</ns3:id> <ns3:name>1000 - Unlimited</ns3:name> <ns3:priceFrom>1000.00</ns3:priceFrom> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:priceBucket> </ns3:getPriceBucketsResponse> </soap:Body> </soap:Envelope> </pre>
--	---

- **getMetadata1 (works the same way for getMetadata2 / getMetadata3)**

	Attributes	Type	Req	Default	Notes
Request	getMetadata1Request	complex	Y		See below for parameters
Response	metadata1	complex			List of metadata1Type elements
Possible Faults	generalError				General error detected. Refer to error details

Metadata1Type complex element contains the following elements

- id - int (metadata1 ID in Price2Spy)
- name – String
- active – boolean
- emailRecipient – String (list of emails where price alert notifications are sent, ‘if ‘Split emails by metadata1’ option is selected in settings)
- dateAdded – datetime (timestamp when the metadata1 was added to Price2Spy)
- dateModified – datetime (timestamp when the metadata1 was modified in Price2Spy)
- addedById – int (ID of user that added the metadata1 into Price2Spy)
- modifiedById – int (ID of user that modified the metadata1 in Price2Spy)

Example SOAP request / response (get all price buckets)

Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getMetadata1Request> <sear:apiKey>88a63363e59afa66266ee8aa69a7aacd6b29c29f</sear:apiKey> </sear:getCategoriesRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <ns3:getMetadata1Response xmlns="http://www.price2spy.com/api/modify" xmlns:ns2="http://www.price2spy.com/api/fault" xmlns:ns3="http://www.price2spy.com/api/search"> <ns3:metatada1> <ns3:id>39</ns3:id> <ns3:name>metadata1 name one</ns3:name> <ns3:active>true</ns3:active> <ns3:emailRecipient>one@domain.com; two@domain.com</ns3:emailRecipient> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> </pre>

	<pre> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:metadata1> <ns3:metadata1> <ns3:id>4142</ns3:id> <ns3:name>metadata1 name two</ns3:name> <ns3:active>true</ns3:active> <ns3:dateAdded>2019-02-14T09:52:38+01:00</ns3:dateAdded> <ns3:dateModified>2019-02-14T10:02:54+01:00</ns3:dateAdded> <ns3:addedById>1234</ns3:addedById> <ns3:modifiedById>1234</ns3:modifiedById> </ns3:metadata1> </ns3:getMetadata1Response> </soap:Body> </soap:Envelope> </pre>
--	---

Matching Suggestions

Introduction

Matching Suggestions (MS) is Price2Spy mechanism for detecting more potential URLs where a particular product can be monitored

List of operations

- Retrieving Matching Suggestions data ('get' operations)
 - i. **getMatchingSuggestions** – get Matching Suggestions for products fulfilling given search criteria
- Managing Matching Suggestions data
 - i. **approveMatchingSuggestions**
 - ii. **rejectMatchingSuggestions**

Operations definition

- **getMatchingSuggestions**

	attributes	Type	Req	Default	Notes
Request search element	searchProducts	complex	Y		See above for searchProductType parameters
Response	List of matchingSuggestion elements	complex			List of matchingSuggestion elements returned by search
Possible Faults	generalError				General error detected. Refer to error details

MatchingSuggestion complex element contains the following elements

- **matchingSuggestionId** –Price2Spy's ID of the Matching Suggestion (will be needed for approveMatchignSuggestion/rejectMatchingSuggestion)
- **myProductId** – Price2Spy's productID of your product for which a Matching Suggestion has been found
- **myProductName** – name of the product for which a Matching Suggestion has been found
- **suggestedUrl** – URL of the Matching Suggestion
- **suggestedUrlSiteHumanName** – Name of the website where Matching Suggestion has been found

- suggestedUrlPrice – Complex element consisting of
 - amount – price captured on the suggested URL
 - currency – currency of the captured price
- suggestedUrlAvailable – boolean (was the product available on the given URL)
 1. Example SOAP request / response (get Matching Suggestions for all products from My Category)
- suggestedUrlName – Name captured on the URL that has been suggested

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getMatchingSuggestionsRequest> <sear:search searchMode="LIKE" categoryName="myCategory"/> <sear:apiKey?</sear:apiKey> </sear:getMatchingSuggestionsRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sear="http://www.price2spy.com/api/search"> <soapenv:Header/> <soapenv:Body> <sear:getMatchingSuggestionsResponse> <!--Zero or more repetitions!--> <sear:matchingSuggestion> <sear:matchingSuggestionId>5309227</sear:matchingSuggestionId> <sear:myProductId>328282</sear:myProductId> <sear:myProductName>My product name</sear:myProductName> <sear:suggestedUrl>http://www.mywebsite.com/xyz</sear:suggestedUrl> <sear:suggestedUrlSiteHumanName>My Website</sear:suggestedUrlSiteHumanName> <sear:suggestedUrlPrice> <sear:amount>33.17</sear:amount> <sear:currency>EUR</sear:currency> </sear:suggestedUrlPrice> <sear:suggestedUrlAvailable>True</sear:suggestedUrlAvailable> <sear:suggestedUrlName>Product 1</sear:suggestedUrlName> </sear:matchingSuggestion> </sear:getMatchingSuggestionsResponse> </soapenv:Body> </soapenv:Envelope></pre>

- **approveMatchingSuggestions**

	Attributes	Type	Req	Default	Notes
Request matchingSug gestionsOp erationType element	matchingSuggestionsOperati onType	complex	Y		See below for matchingSuggestionsOper ationType parameters
Response	List of matchingSuggestionsAppr oveResponse elements	complex			List of matchingSuggestionsAppr oveResponse elements returned by search
Possible Faults	generalError				General error detected. Refer to error details

MatchingSuggestionOperationType - complex element contains the following elements

- productID – Price2Spy’s ID of the product for which we’re approving / rejecting the Matching Suggestion

- matchingSuggestionId – ID of the Matching Suggestion we're approving / rejecting (this element can be repeated N times)

MatchingSuggestionApproveResponse - complex element contains one or more sequence of the following elements

- urlId – URL_ID of the URL created by approving the Matching Suggestion

Example SOAP request / response (Approve 2 Matching Suggestions for a particular product)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:matchingSuggestionsApproveRequest> <mod:matchingSuggestion productId="5732891"> <mod:matchingSuggestionId>188363661</mod:matchingSuggestionId> <mod:matchingSuggestionId>188363662</mod:matchingSuggestionId> </mod:matchingSuggestion> <mod:apiKey>?</mod:apiKey> </mod:matchingSuggestionsApproveRequest> </soapenv:Body> </soapenv:Envelope></pre>
Response	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:matchingSuggestionsApproveResponse> <mod:urlId>1723890</mod:urlId> <mod:urlId>1723891</mod:urlId> </mod:matchingSuggestionsApproveResponse> </soapenv:Body> </soapenv:Envelope></pre>

- **rejectMatchingSuggestions**

	Attributes	Type	Req	Default	Notes
Request matchingSuggestionOperationType element	matchingSuggestionOperationType	complex	Y		See below for matchingSuggestionOperationType parameters
Response					(void) – the presence of this element indicates that deleteUrl) was successfully carried out (otherwise a fault would be raised)
Possible Faults	generalError				General error detected. Refer to error details

Example SOAP request / response (Reject 2 Matching Suggestions for a particular product)

Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body> <mod:matchingSuggestionsRejectRequest> <mod:matchingSuggestion productId="5732891"> <!--1 or more repetitions:--></pre>
----------------	--

	<pre> <mod:matchingSuggestionId>188363663</mod:matchingSuggestionId> <mod:matchingSuggestionId>188363664</mod:matchingSuggestionId> </mod:matchingSuggestion> <mod:apiKey>?</mod:apiKey> </mod:matchingSuggestionsRejectRequest> </soapenv:Body> </soapenv:Envelope> </pre>
Response	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mod="http://www.price2spy.com/api/modify"> <soapenv:Header/> <soapenv:Body/> </soapenv:Envelope> </pre>

Implementation of Price2Spy API Clients - SOAP

PHP

You can use our implementation of PHP Client for Price2Spy API, or you can build your client using our WSDL. Here are basic examples of our PHP client.

PHP code samples

Below is an example of how to initialize Price2Spy PHP Client.

```

require_once '../sources/p2sApiService.php';

$wsdl_location = 'https://api.price2spy.com/soap?wsdl';
$p2sApi = new p2sApiService(array(), $wsdl_location);

```

In case of HTTPS-related connection problems

Certain versions of PHP require the following SSL handshake protocols to be defined

```

require_once '../sources/p2sApiService.php';

$context = stream_context_create([
    'ssl' => [
        'ciphers' => 'DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:AES256-
SHA:KRB5-DES-CBC3-MD5:KRB5-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-
SHA:DES-CBC3-SHA:DES-CBC3-MD5:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:AES128-
SHA:RC2-CBC-MD5:KRB5-RC4-MD5:KRB5-RC4-SHA:RC4-MD5:RC4-SHA:RC4-MD5:KRB5-DES-CBC-
MD5:KRB5-DES-CBC-SHA:EDH-RSA-DES-CBC-SHA:EDH-DSS-DES-CBC-SHA:DES-CBC-SHA:DES-CBC-
MD5:EXP-KRB5-RC2-CBC-MD5:EXP-KRB5-DES-CBC-MD5:EXP-KRB5-RC2-CBC-SHA:EXP-KRB5-DES-
CBC-SHA:EXP-EDH-RSA-DES-CBC-SHA:EXP-EDH-DSS-DES-CBC-SHA:EXP-DES-CBC-SHA:EXP-RC2-
CBC-MD5:EXP-RC2-CBC-MD5:EXP-KRB5-RC4-MD5:EXP-KRB5-RC4-SHA:EXP-RC4-MD5:EXP-RC4-
MD5',
    ],
]);

$wsdl_location = 'https://api.price2spy.com/soap?wsdl';
$p2sApi = new p2sApiService(array('stream_context' => $context), $wsdl_location);

```

2. deleteProductRequestType

```

/deleteProductRequestType = new deleteProductRequestType();
/deleteProductRequestType->apiKey = 'xyz';
/deleteProductRequestType->productId = 149197;

$p2sApi->deleteProduct($deleteProductRequestType);

```


6. getUrls

```
$getUrlsRequestType = new getUrlsRequestType();
$getUrlsRequestType->apiKey = 'xyz';
$getUrlsRequestType->search = new searchUrlsType();
$getUrlsRequestType->search->id = '7112';
$getUrlsRequestType->search->idMode = 'ID';

$urls = $p2sApi->getUrls($getUrlsRequestType);

echo 'Found '.count($urls->url).' urls:<br/><hr/>';
if($urls->url != null) {
    foreach($urls->url as $url) {
        if($url->lastMeasurement != null) {
            echo $url->siteHumanName . ': ' . $url->lastMeasurement->price->amount . ' ' .
            $url->lastMeasurement->price->currency . ' ' . '<br/>';
        } else {
            echo $url->siteHumanName . ': no measurements.<br/>';
        }
    }
}
```

7. insertProduct

```
$insertProductRequestType = new insertProductRequestType();
$insertProductRequestType->apiKey = 'xyz';
$insertProductRequestType->insert = new insertProductType();
$insertProductRequestType->insert->productName = 'Insert API Product';
$insertProductRequestType->insert->checkFrequencyType = 'DAYS';
$insertProductRequestType->insert->checkFrequencyInterval = 1;
$insertProductRequestType->insert->active = true;
$insertProductRequestType->insert->sku = 'SKU';
$insertProductRequestType->insert->internalId = 'InternalId';
$insertProductRequestType->insert->targetPrice = 12.34;
$insertProductRequestType->insert->purchasePrice = 56.78;
$insertProductRequestType->insert->categoryName = 'New Category';
$insertProductRequestType->insert->brandName = 'Existing Brand';
$insertProductRequestType->insert->supplierName = '';

$id = $p2sApi->insertProduct($insertProductRequestType);

echo 'New product id: '.$id.<br/>';
```

8. insertUrl

```
$insertUrlRequestType = new insertUrlRequestType();
$insertUrlRequestType->apiKey = 'xyz';
$insertUrlRequestType->insert = new insertUrlType();
$insertUrlRequestType->insert->productId = 149198;
$insertUrlRequestType->insert->url =
'http://www.amazon.com/gp/product/B0099QRVZS/ref=s9_simh_gw_p107_d0_i3?pf_rd_m=ATVPDKIKX0DER
&pf_rd_s=center-
2&pf_rd_r=009RBEJERGQ2217R4HHQ&pf_rd_t=101&pf_rd_p=1389517282&pf_rd_i=507846';
$insertUrlRequestType->insert->active = true;
$insertUrlRequestType->insert->showInReports = true;

$id = $p2sApi->insertUrl($insertUrlRequestType);

echo 'New urls id: '.$id.<br/>';
```

9. updateProduct

```
$updateProductRequestType = new updateProductRequestType();
$updateProductRequestType->apiKey = 'xyz';
$updateProductRequestType->update = new updateProductType();
$updateProductRequestType->update->productId = 149198;
$updateProductRequestType->update->productName = 'New API Product Name';
$updateProductRequestType->update->active = false;
```



```
$p2sApi->updateProduct($updateProductRequestType);
```

10. updateUrl

```
$updateUrlRequestType = new updateUrlRequestType();
$updateUrlRequestType->apiKey = 'xyz';
$updateUrlRequestType->update = new updateUrlType();
$updateUrlRequestType->update->urlId = 293235;
$updateUrlRequestType->update->url = 'http://www.amazon.com/Samsung-Factory-Unlocked-
International-Version/dp/B009JJT4KK/ref=pd_sim_cps_1';
$updateUrlRequestType->update->counterFailed = 0;
$updateUrlRequestType->update->showInReports = false;

$p2sApi->updateUrl($updateUrlRequestType);
```

11. getBrands

```
$getBrandsRequestType = new getBrandsRequestType();
$getBrandsRequestType ->apiKey = 'xyz';

$brands = $p2sApi->getBrands ($ getBrandsRequestType );

echo 'Found '.count($brands->brand).' brands:<br/><hr/>';
if($brands->brand != null) {
    foreach($brands->brand as $brand) {
        echo $brand->name . '<br/>';
    }
}
```

12. getCategories

```
$getCategoriesRequestType = new getCategoriesRequestType();
$getCategoriesRequestType->apiKey = 'xyz'

$categories = $p2sApi->getCategories($getCategoriesRequestType);

echo 'Found '.count($categories->category).' categories:<br/><hr/>';
if($categories->category != null) {
    foreach($categories->category as $category) {
        echo $category->name . '<br/>';
    }
}
```

13. getSuppliers

```
$getSuppliersRequestType = new getSuppliersRequestType();
$getSuppliersRequestType ->apiKey = 'xyz'

$suppliers = $p2sApi->getSuppliers($getSuppliersRequestType);

echo 'Found '.count($suppliers->supplier).' suppliers:<br/><hr/>';
if($suppliers->supplier != null) {
    foreach($suppliers->supplier as $supplier) {
        echo $supplier->name . '<br/>';
    }
}
```

14. getPriceBuckets

```
$getPriceBucketsRequestType = new getPriceBucketsRequestType();
$getPriceBucketsRequestType->apiKey = $apiKey;

$priceBuckets = $p2sApi->getPriceBuckets($getPriceBucketsRequestType);
```

```
echo 'Found '.count($priceBuckets->priceBucket).' price buckets:<br/><hr/>';
if($priceBuckets->priceBucket != null) {
    foreach($priceBuckets->priceBucket as $priceBucket) {
        echo $priceBucket->name . '<br/>';
    }
}
```