

Chapter 2 - Discovery - Testing access to the web service with the browser

Discovery - Testing access to the web service with the browser

To test if you have properly configured your access to the web service, access your online shop with the following URL: `http://mypasskey@example.com/api/`, where `mypasskey` is replaced with the key you created, and `example.com` with your shop's URL. Mozilla Firefox is the preferred browser to test access, but any browser able to display XML should do just fine.

The shop should prompt you for a username and a password to enter. The ID is the authentication key and **there is no password**.



Another method is to go directly to the following page of your shop: `http://example.com/api/`

Of course, it can also work locally: `http://127.0.0.1/ps1541/api/` gives the same result.

The `/api/` URL gives you access to the list of resources that you configured in your back-office, with all the permissions you chose to grant. **If no permission has been set for the key, than the browser will keep asking you to enter the key indefinitely.** If you cannot access some resources, the API might answer with this XML response:

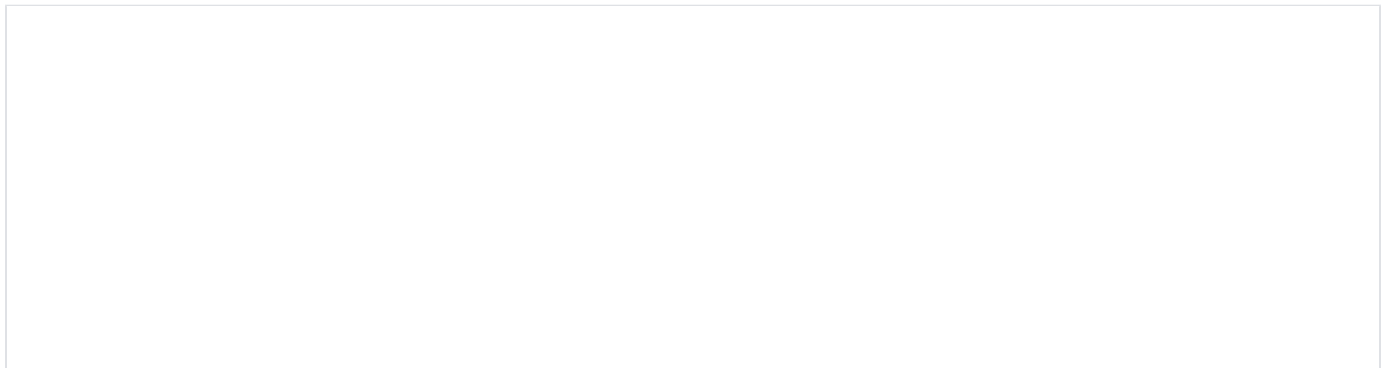
```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <errors>
    <error>
      <message><![CDATA[Internal error. To see this error please display the PHP errors.]]></message>
    </error>
  </errors>
</prestashop>
```

Using XLink, you will then be able to access your various resources. XLink associates an XML file to another XML file via a link.

Available resources

The `/api/` URL gives you the root of all the resources, in the form of an XML file.

Here it is, extremely simplified. This list is current as of version 1.5.4.1 of PrestaShop. Note that we only show the API resources available for one of the available stores.



```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <api shop_name="MYSHOP">
    <addresses>...</addresses>
    <carriers>...</carriers>
    <cart_rules>...</cart_rules>
    <carts>...</carts>
    <categories>...</categories>
    <combinations>...</combinations>
    <configurations>...</configurations>
    <contacts>...</contacts>
    <content_management_system>...</content_management_system>
    <countries>...</countries>
    <currencies>...</currencies>
    <customer_messages>...</customer_messages>
    <customer_threads>...</customer_threads>
    <customers>...</customers>
    <deliveries>...</deliveries>
    <employees>...</employees>
    <groups>...</groups>
    <guests>...</guests>
    <image_types>...</image_types>
    <images>...</images>
    <languages>...</languages>
    <manufacturers>...</manufacturers>
    <order_carriers>...</order_carriers>
    <order_details>...</order_details>
    <order_discounts>...</order_discounts>
    <order_histories>...</order_histories>
    <order_invoices>...</order_invoices>
    <order_payments>...</order_payments>
    <order_states>...</order_states>
    <orders>...</orders>
    <price_ranges>...</price_ranges>
    <product_feature_values>...</product_feature_values>
    <product_features>...</product_features>
    <product_option_values>...</product_option_values>
    <product_options>...</product_options>
    <product_suppliers>...</product_suppliers>
    <products>...</products>
    <search >...</search>
    <shop_groups>...</shop_groups>
    <shops>...</shops>
    <specific_price_rules>...</specific_price_rules>
    <specific_prices>...</specific_prices>
    <states>...</states>
    <stock_availables>...</stock_availables>
    <stock_movement_reasons>...</stock_movement_reasons>
    <stock_movements>...</stock_movements>
    <stocks>...</stocks>
    <stores>...</stores>
    <suppliers>...</suppliers>
    <supply_order_details>...</supply_order_details>
    <supply_order_histories>...</supply_order_histories>
    <supply_order_receipt_histories>...</supply_order_receipt_histories>
    <supply_order_states>...</supply_order_states>
    <supply_orders>...</supply_orders>
    <tags>...</tags>
    <tax_rule_groups>...</tax_rule_groups>
    <tax_rules>...</tax_rules>
    <taxes>...</taxes>
    <translated_configurations>...</translated_configurations>
    <warehouse_product_locations>...</warehouse_product_locations>
    <warehouses>...</warehouses>
    <weight_ranges>...</weight_ranges>
    <zones>...</zones>
  </api>
  <api shop_name="MYOTHERSHOP">...</api>
  <api shop_name="YETANOTHERSHOP">...</api>
</prestashop>
```

As with any XLink-bearing XML file, each resource element leads to more resources, linked from the `xlink` attribute.

```
<customers xlink:href="http://example.com/api/customers" get="true" put="true" post="true" delete="true" head="true">
  <description xlink:href="http://example.com/api/customers" get="true" put="true" post="true" delete="true" head="true">The e-shop's customers</description>
  <schema xlink:href="http://example.com/api/customers?schema=blank" type="blank"/>
  <schema xlink:href="http://example.com/api/customers?schema=synopsis" type="synopsis"/>
</customers>
```

In addition, the element contains a description of the resource, and two schemas: a blank one, which you can use to create a new item, and a synopsis one, which is just like the blank one but with a detailed description of what type of data is expected in each element.

Here is an extract of the Customer blank schema:

Blank schema

```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <customer>
    <id></id>
    <id_default_group></id_default_group>
    <id_lang></id_lang>
    <newsletter_date_add></newsletter_date_add>
    <ip_registration_newsletter></ip_registration_newsletter>
    <last_passwd_gen></last_passwd_gen>
    <secure_key></secure_key>
    <deleted></deleted>
    <passwd></passwd>
    <lastname></lastname>
    <firstname></firstname>
    <email></email>
    ...
  </customer>
</prestashop>
```

Here is an extract of the Customer synopsis schema:

Synopsis schema

```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <customer>
    <id_default_group></id_default_group>
    <id_lang format="isUnsignedId"></id_lang>
    <newsletter_date_add></newsletter_date_add>
    <ip_registration_newsletter></ip_registration_newsletter>
    <last_passwd_gen></last_passwd_gen>
    <secure_key format="isMd5"></secure_key>
    <deleted format="isBool"></deleted>
    <passwd required="true" maxSize="32" format="isPasswd"></passwd>
    <lastname required="true" maxSize="32" format="isName"></lastname>
    <firstname required="true" maxSize="32" format="isName"></firstname>
    <email required="true" maxSize="128" format="isEmail"></email>
    ...
  </customer>
</prestashop>
```

The value types can be found in the [Web service reference](#).

Navigating your data

In this tutorial, we are going to base our examples on the management of customers through the API, but you can use just about any data

In the Customers element, you should get these attributes and tags:

```
<customers xlink:href="http://example.com/api/customers" get="true" put="true" post="true" delete="true" head="true">
  <description xlink:href="http://example.com/api/customers" get="true" put="true" post="true" delete="true" head="true">The e-shop's customers</description>
  <schema xlink:href="http://example.com/api/customers?schema=blank" type="blank"/>
  <schema xlink:href="http://example.com/api/customers?schema=synopsis" type="synopsis"/>
</customers>
```

The get, put, post, and delete attributes have the value true, meaning that you have correctly configured the customers resource for the current, and that their data is accessible.

You can now use the XLink link that shows up on the URL `http://example.com/api/customers` and go to it:

Result for <http://example.com/api/customers>

```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <customers>
    <customer id="1" xlink:href="http://example.com/api/customers/1"/>
    <customer id="2" xlink:href="http://example.com/api/customers/2"/>
    <customer id="3" xlink:href="http://example.com/api/customers/3"/>
    <customer id="4" xlink:href="http://example.com/api/customers/4"/>
    <customer id="5" xlink:href="http://example.com/api/customers/5"/>
    <customer id="6" xlink:href="http://example.com/api/customers/6"/>
    <customer id="7" xlink:href="http://example.com/api/customers/7"/>
    ...
  </customers>
</prestashop>
```

Then, from the customers list which is displayed via `http://example.com/store/api/customers`, you can access the XLink corresponding to each customer. This gives you all the information

Result for <http://example.com/api/customers/1>

```
<?xml version="1.0" encoding="UTF-8"?>
<prestashop xmlns:xlink="http://www.w3.org/1999/xlink">
  <customer>
    <id><![CDATA[1]]></id>
    <id_default_group xlink:href="http://example.com/api/groups/3"><![CDATA[3]]></id_default_group>
    <id_lang xlink:href="http://example.com/api/languages/1"><![CDATA[1]]></id_lang>
    <newsletter_date_add><![CDATA[2013-04-23 11:40:52]]></newsletter_date_add>
    <ip_registration_newsletter></ip_registration_newsletter>
    <last_passwd_gen><![CDATA[2013-04-23 05:40:52]]></last_passwd_gen>
    <secure_key><![CDATA[5b402973df21cd42a303a2fdfce91df7]]></secure_key>
    <deleted><![CDATA[0]]></deleted>
    <passwd><![CDATA[a74796131824211b33ed551dcff10a9]]></passwd>
    <lastname><![CDATA[DOE]]></lastname>
    <firstname><![CDATA[John]]></firstname>
    <email><![CDATA[pub@prestashop.com]]></email>
    <id_gender><![CDATA[1]]></id_gender>
    <birthday><![CDATA[1970-01-15]]></birthday>
    <newsletter><![CDATA[1]]></newsletter>
    <optin><![CDATA[1]]></optin>
    <website></website>
    <company></company>
    <siret></siret>
    <ape></ape>
    <outstanding_allow_amount><![CDATA[0.000000]]></outstanding_allow_amount>
    <show_public_prices><![CDATA[0]]></show_public_prices>
    <id_risk><![CDATA[0]]></id_risk>
    <max_payment_days><![CDATA[0]]></max_payment_days>
    <active><![CDATA[1]]></active>
    <note></note>
    <is_guest><![CDATA[0]]></is_guest>
    <id_shop><![CDATA[1]]></id_shop>
    <id_shop_group><![CDATA[1]]></id_shop_group>
    <date_add><![CDATA[2013-04-23 11:40:52]]></date_add>
    <date_upd><![CDATA[2013-04-23 11:40:52]]></date_upd>
    <associations>
      <groups node_type="group">
        <group xlink:href="http://example.com/api/groups/3">
          <id><![CDATA[3]]></id>
        </group>
      </groups>
    </associations>
  </customer>
</prestashop>
```

Adding and editing a resource

The <http://example.com/api/customers/1> example is not only available using the HTTP GET method, but also using POST, PUT, DELETE, HEAD.

To add a customer (or any other resource, for that matter), you simply need to GET the XML blank data for the resource (`/api/customer?schema=blank`), fill it with your changes, and POST the whole XML file to the `/api/customers/` URL again. PrestaShop will take care of adding everything in the database, and will return an XML file indicating that the operation has been successful, along with the ID of the newly created customer.

To edit an existing resource: GET the full XML file for the resource you want to change (`/api/customers/7`), edit its content as needed, then PUT the whole XML file back to the same URL again.