

Chapter 6 - Data creation - Creating a remote online form

Chapter 6 - Data creation: Creating a remote online form

Creation - Remote Online Form

Objective: A Web application to list and create a new customer.

Difficulty: **

Preparation

Copy the file `list_the_customers.php` from Section 3.3 to a file named `C-CRUD.php` at the root of your Web server.

The addition of resource can be likened to an upgrade from an empty element.

But how do we retrieve an XML formatted as an empty customer?

In the web service, there is a method to retrieve an empty XML. It is accessible via a URL formatted as follows: [http://example.com/api/\(resource name\)?schema=blank](http://example.com/api/(resource name)?schema=blank)

It is possible to replace the parameter scheme value "blank" with "synopsis" in order to gain more information about the resource fields.

As we saw in Section 3.3 (List customers) it is possible make an array of parameters for "get", "resource," and "id." It is also possible to specify only one URL this way:

```
$xml = $webService->get(array('url' => 'http://example.com/api/customers?schema=blank'));
```

Here, we get the entire XML variable from an empty customer.

```
<prestashop>
<customer>
  <id_default_group/>
etc...Beginning of the XML file retrieved:
```

We can then, thanks to the many fields we have, create an associated form.

Getting of all fields

```
$resources = $xml->children()->children();
```

Path of all fields and part of the dynamic creation of form fields in a table

```
foreach ($resources as $key => $resource) {
  echo '<tr><th>' . $key . '</th><td>';
  echo '<input type="text" name="' . $key . '" value="" />';
  echo '</td></tr>';
}
```

Once the data is passed in POST, we combined the data sent with the blank XML file, which is the same technique used for updating data.

```
foreach ($resources as $nodeKey => $node) {
    $resources->$nodeKey = $_POST[$nodeKey];
}
```

Calling the web service is similar to what we have seen previously:

```
$opt = array('resource' => 'customers');
$opt['postXml'] = $xml->asXML();
$xml = $webService->add($opt);
```

Now create a script that adds a customer. Remember that some fields are mandatory, so do not forget to fill them out.

If you have trouble, look at the code the `3-Create.php` sample file.

When a customer is created from within PrestaShop's administration interface, a confirmation e-mail is sent to the customer. This cannot be done directly with the webservice: there is no way to trigger the sending of that confirmation e-mail.

However, you can create an override file for the `Customer` class and override the `addWs()` method. This method is similar to `ObjectModel::add()` but is only called from the webservice. You can find examples of its use in the `Product` and `Order` classes.