

# Chapter 4 - Retrieve Data - Retrieving a Client

## Retrieve Data - Retrieving a Client

**Objective:** A Web application to list and display information from a customer

**Difficulty:** \*

**Problem:** How to create a system that allows customers using IDs to retrieve customer records?

### Preparation

Duplicate the file `list_the_clients.php` from the previous step to a file named `R-CRUD.php` at the root of your Web server.

If you didn't succeed at the previous step, duplicate the file `0-CustomersList.php` to a file named `R-CRUD.php`.

In the XML file containing the list of clients, we will find all the XLinks providing access to customer information.

```
<customers>
  <customer id="1" xlink:href="http://example.com/store/api/customers/1" />
</customers>
```

### Example

Here we see that the xlink for the "customer" tag with ID 1 is: <http://mystore.com/api/customers/1>

This link leads to an XML file containing information about the client with ID 1.

For this tutorial, in order to manage access to different clients you will proceed by associating page identifiers with customers via a GET parameter named "id".

The <http://mystore.com/R-CRUD.php?id=1> we will display the customer 1's file.

Modify the table created in the previous chapter to add a link to future customer files.

You'll have to isolate the display from the display list of a particular customer.

In order to do this you must isolate the display of your list by verifying, using `isset()`, that the GET "id" parameter property is not present when viewing your list.

Calling the web service is exactly the same as for displaying the list, except for if you need to add the 'id' element to the table whose value is the id of a client.

At this moment, we are using "customers" or "client" resources. If we'd been trying to change the "countries" resources, this id would have been a country id.

```
$opt[ 'resource' ] = 'customers';
$opt[ 'id' ] = $_GET[ 'id' ];
$xml = $webService->get( $opt );
```



Use `isset()` before setting an ID allows you to easily carry out everything in this chapter.

Accessing resources is performed as above for displaying the list because the tags that interest us are children of the "customers" tag.

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

This path is done in another way (here in an HTML table):

```
foreach ( $resources as $key => $resource )  
    echo 'Name of field: ' . $key . ' - Value: ' . $resource . '<br />';
```

You now have everything needed to create a script to both list and display information for a particular client.

Try creating this script `R-CRUD.php`. If you encounter any difficulties, follow the example from the file `1-Retrieve.php` which corresponds to the result you should get.

We will see in another tutorial how to filter, sort and limit the number of items displayed in the list.

If you're in a hurry to implement these features, you can find more information in Chapter 8.