3.2 - Handling errors

Handling errors

Learning error handling with the library is essential to begin. If you implement this verification directly, you immediately detect where the error comes from along with other information.

Error handling with the PHP library from the web service is done with the help of exceptions.

The principle

The treatments related to the PrestaShop web service must be within a try block which itself must be followed by a catch block to retrieve the errors and, if possible, to catch them.

Illustration:

```
try {
  // Execution (stops and goes in the catch block if an error occurs)
}
catch {
  // Error handling (tries to catch the error or the error display)
}
```

Example

```
try {
  // creating web service access
  $webService = new PrestaShopWebservice( 'http://maboutique.com/', 'ZR92FNY5UFRERNI3O9Z5QDHWKTP3YIIT', false );
  // call to retrieve all clients
  $xml = $webService->get( array( 'resource' => 'customers' ) );
}
catch ( PrestaShopWebserviceException $ex ) {
  $trace = $ex->getTrace(); // Retrieve all information on the error
  $errorCode = $trace[ 0 ][ 'args' ][ 0 ]; // Retrieve the error code
  if ( $errorCode == 401 )
    echo 'Bad auth key';
  else
    echo 'Other error : <br />' . $ex->getMessage(); // Shows a message related to the error
}
```

That means each creation or use of the library must be located within a "try" block. The "catch" block can then handle the error if it occurs during the execution of the try block.
Now we'll see how to list all customers via the web service, and then we will see the four CRUD methods.