ex01_SoaGatewayFirst.php

```php
<?php

/*
 *      This set of examples demonstrate the simplicity in accessing Adabas
 *      data as "WebServices" from PHP, based on the "EmployeesMini" view
 *      representing a subset ("View") of the Adabas demo file "Employees".
 *
 *      This first example outlines the usage of the PHP SoapClient class,
 *      which provides the infrastructure for issuing SOAP requests.
 *
 *      All examples are based on an Adabas SOA Gateway server running on
 *      host "soagate", port 8082, these need to be adjusted to the actual
 *      server host/port used at your site.
 *
 */


/*
 *      Instantiate the PHP "SoapClient" class
 */
try {
        /*
         * The only required parameter when instantiating "SoapClient" is the URL
         * pointing to the WSDL for the WebService to be accessed.
         */
        $soapclient = new SoapClient("http://localhost:8022/adabas_EmployeesMini?WSDL");
} catch (SoapFault $soapfault) {
        /*
         * In case of a SOAPFault being thrown, this will be caught and the fault
         * information printed. If a SOAPFault occurs outside of a "try/catch"
         * structure, PHP will abend the script with a generic message.
         *
         * Here we print the SOAPFault information in a structured way, the "<pre>"
         * tags are required to format the object nicely.
         */
        echo "<pre>";
        print_r($soapfault);
        echo "</pre>";
        return;
}

/*
 *      Without any knowledge of, or looking at Adabas definitions, a "WebService programmer"
 *      can easily retrieve the signature of any exposed Adabas SOA Gateway Service and
 *      code based on it.
 *
 *      First we use a method of the SoapClient class to retrieve the functions exposed
 *      by the specific "WebService", and print it:
 */

echo "<pre><B>Functions:</B>\n\n";
print_r($soapclient->__getFunctions());

/*
 *      The function prototypes shown by the __getFunction() method also depict the required
 *      parameters and the return values. Their definitions can be printed as well:
 */
echo "\n\n-------------------------------------------------------------------\n\n<B>Type Definitions:</B>\n\n";
print_r($soapclient->__getTypes());
echo "</pre>";

/*
 *      Proceed to ex02_SoaGatewayEmpList.php - List and format Employees data
 */

?>
```