**ijnWebService**

**Cómo utilizar nuestra librería?**

• Coloque los certificados generados previamente por Ud. (.crt y .key) dentro del directorio:  
 - /TA\_Produccion (los certificados generados para Producción)  
 - /TA\_Test (los certificados generados para Test “Homologación”)

• Edite el archivo configuración.ini para indicar los nombres de los certificados generados previamente por Ud. (.crt y .key)

• Incluya nuestra librería en su proyecto: **require 'ijnwebservice.php';**

**Inicialización para utilizar los WebServices de la AFIP:**

Nuestra clase se puede inicializar de dos maneras: Para Homologación o Producción.

**$webservice = new ijnWebService(CUIT, modo);**

Como observamos en el constructor hay dos parametros:

• CUIT: Es el cuit sin guiones correspondiente al certificado generado en la afip.

• modo: Indica que el webservice se inicializa en modo Homologación (test) o Producción  
 - Para Homologación (test): “T”  
 - Para Producción: “P”

Ejemplo:  
 Para Test: $webservice = new ijnWebService(“11111111111”, “T”);  
 Para Producción: $webservice = new ijnWebService(“11111111111”, “P”);

**Manejo de Errores:**

Ud. Podrá a la hora de generar o consultar comprobantes manejar los errores, en caso que los haya, atraves del array “errores”.

***Descripción del array:***  
errores["ErrorCodigo"] – Devuelve el código de error  
errores["ErrorMensaje"] – Devuelve el mensaje del error

**Métodos disponibles:**

**Crear y Asignar CAE a un comprobante: ( ->crearComprobante() )**Se emplea el método crearComprobante(). Para emplear este método, antes de llamarlo se deben completar los siguientes parámetros:

Código:

<?php  
require 'ijnwebservice.php';  
  
$webservice = new ijnWebService("11111111111","T"); //inicializamos en modo test  
$webservice->factura->cantidadRegistros="1"; // Cantidad de comprobantes a registrar  
$webservice->factura->puntoVenta="2"; // Punto de venta  
$webservice->factura->codigoComprobante="1"; // Código del comprobante a generar  
$webservice->factura->codigoConcepto="1"; // 1:Productos 2:Servicios 3:Productos y Servicios  
$webservice->factura->codigoDocumento="80"; //Tipo de documento 80-CUIT  
$webservice->factura->numeroDocumento="27066479701"; //Numero de CUIT o Documento del comprador  
$webservice->factura->numeroComprobante=”1”; // Número del comprobante  
$webservice->factura->fecha="20201022"; // Fecha del comprobante  
$webservice->factura->importeTotal="121"; // Importe total del comprobante  
$webservice->factura->importeNoGravado="0"; // Importe neto no gravado  
$webservice->factura->importeNeto="100"; // Importe neto gravado  
$webservice->factura->importeExento="0"; // Importe exento de IVA  
$webservice->factura->importeOtrosTributos="0"; //Importe total de tributos  
$webservice->factura->importeIVA="21"; //"Solamente para comprobantes que no sean tipo C"  
$webservice->factura->FchServDesde="";  
$webservice->factura->FchServHasta="";  
$webservice->factura->FchVtoPago="";  
$webservice->factura->codigoMoneda="PES"; //Tipo de moneda usada  
$webservice->factura->cotizacionMoneda="1"; // Cotización de la moneda usada (1 para pesos argentinos)  
  
//Agregar Alicuotas de IVA "Solamente para comprobantes que no sean tipo C"  
$webservice->factura->ivacod="5";  
$webservice->factura->ivabase="100";  
$webservice->factura->ivaimp="21";  
$webservice->factura->agregarIvas(1);  
  
//Agregar Tributos en caso que sea necesario  
$webservice->factura->tribId="99";  
$webservice->factura->tribDesc="Ganancias";  
$webservice->factura->tribBaseImp="100"; //la Base imponible es el total de los importes netos de la factura  
$webservice->factura->tribAlic="5";  
$webservice->factura->tribImporte="5";  
$webservice->factura->agregarTributos(1);  
  
//Por ultimo, llamamos al metodo para generar el comprobante  
**$cmp=$webservice->crearComprobante();**  
  
//Comprobamos si el comprobante fue generado correctamente  
if($cmp!=null){  
 echo "CAE: ".$webservice->comprobante["CAE"]."<br>";  
 echo "CAE Vencimiento: ".$webservice->comprobante["CAEVto"]."<br>";  
 echo "Resultado: ".$webservice->comprobante["Resultado"]."<br>";  
} else {  
 echo "Resultado: ".$webservice->comprobante["Resultado"]."<br>";  
 echo "ErrCode: ".$webservice->errores["ErrorCodigo"]."<br>";  
 echo "ErrorMensaje: ".$webservice->errores["ErrorMensaje"]."<br>";  
}

?>

**Descripción del Código:**

Como vemos, al generar el comprobante, la variable **$cmp** va a contener la respuesta de la afip en formato XML.

***Si el comprobante fue Aprobado por afip:***  
En el caso que haya sido **aprobado** por AFIP, el **array “comprobante”** va a contener lo siguiente:

- comprobante["Resultado"] Contiene la aprobación de la afip. Si fue aprobado sin errores contiene “A”  
- comprobante["CAE"] Contiene el CAE del comprobante  
- comprobante["CAEVto"] Contiene el vencimiento del CAE

***Si el comprobante fue Rechazado por afip:***  
En el caso que haya sido **rechazado** por AFIP, el **array “comprobante”** va a contener lo siguiente:

- comprobante["Resultado"] Contiene el rechazo de la afip. Si fue rechazado contiene “R”  
- comprobante["CAE"] El CAE es nulo  
- comprobante["CAEVto"] El vencimiento del CAE es nulo

Y el **array “errores”** va a contener el código de error y su mensaje:

- errores["ErrorCodigo"] Contiene el codigo de error  
- errores["ErrorMensaje"] Contiene el mensaje de error

De esta manera, podemos asegurarnos que el comprobante fue autorizado correctamente por la AFIP. Por este motivo realizamos un chequeo luego de generar el comprobante para que la variable **$cmp** no sea nula, como apreciamos al final del código.

**Con respecto a las “Alícuotas de IVA”:**

Las alicuotas del IVA solamente se agregaran en el caso que los comprobantes a generar **no sean tipo “C”.**  
En el caso que los comprobantes a generar sean tipo “A”, “B” ó “M”, Sí se deben agregar las alicuotas de iva como vemos en el codigo:

$webservice->factura->ivacod="5"; // código del iva correspondiente al 21%  
$webservice->factura->ivabase="100"; // base imponible  
$webservice->factura->ivaimp="21"; // valor del iva  
$webservice->factura->agregarIvas(1); // metodo para agregar el iva

En este caso, estamos agregando una alicuota de IVA del 21% (código 5) sobre una base imponible de 100.  
En el caso de querer agregar otra alícuota de IVA:

$webservice->factura->ivacod="4"; // código del iva correspondiente al 10.5%  
$webservice->factura->ivabase="100"; // base imponible  
$webservice->factura->ivaimp="10.5"; // valor del iva  
$webservice->factura->agregarIvas(2); // metodo para agregar el iva

Como observamos, completamos los campos correspondientes y en el metodo factura->agregarivas(2) ponemos “2” como parametro para indicar que se agrega la segunda alicuota. En este caso estamos agregando también una alícuota del 10.5% con su correspondiente código “4”, sobre una base imponible de 100 y un valor de iva de 10.5.

***Para tener en cuenta:***   
Si el comprobante que se genera contiene un solo valor de IVA, por ejemplo si se facturan productos todos al 21%, se debe agregar sólo Una Alícuota de Iva, en este caso del 21%.

**Con respecto a los “Tributos”:**

Los Tributos se agregan, en el caso que existan en el comprobante a generar, como se indica en el codigo expuesto.  
  
$webservice->factura->tribId="99";  
$webservice->factura->tribDesc="Ganancias";  
$webservice->factura->tribBaseImp="100"; //la Base imponible es el total de los importes netos de la factura  
$webservice->factura->tribAlic="5";  
$webservice->factura->tribImporte="5";  
$webservice->factura->agregarTributos(1);

Como podemos apreciar, los tributos se agregan de la misma manera que las alícuotas del iva. En caso de agregar otro tributo, quedaría:

$webservice->factura->tribId="90";  
$webservice->factura->tribDesc="Impuestos Internos";  
$webservice->factura->tribBaseImp="100"; //la Base imponible es el total de los importes netos de la factura  
$webservice->factura->tribAlic="2";  
$webservice->factura->tribImporte="2";  
$webservice->factura->agregarTributos(2);

**Con respecto a los “Comprobantes Asociados”:**

Los comprobantes asociados para informar deben ser Comprobantes “A”, “B”, “C”, “M” ya generados anteriormente. Los Remitos no se deben informar en el webservice como comprobante asociado.  
Por ejemplo si estamos generando una Nota de Credito “A” para anular la factura “A” Nº 00002-00000026, como comprobante asociado a la Nota de Crédito se debe indicar dicha factura.

$webservice->factura->compAsocTipo="1"; // código de comprobante Factura A  
$webservice->factura->compAsocPtoVta="2"; // punto de venta de dicha factura A  
$webservice->factura->compAsocNro="26"; // número de dicha factura A  
$webservice->factura->agregarCompAsoc(1);

De esta manera, estamos agregando según el ejemplo citado, como comprobante asociado a la Nota de Credito “A” que estamos generando, la factura “A” Nº 00002-00000026. Por ejemplo, para anularla.

***Nota Importante:***El comprobante asociado debe de existir SÍ o SÍ en los servidores de AFIP, es decir, debieron de haberse generado en algún momento. En caso contrario, la afip responde con un error, consultando el mismo desde el array “errores” comentado anteriormente.

**Método para Consultar el Número del Último comprobante generado:**

Es muy importante saber el número del último comprobante generado. El método es:

->**UltimoCmp(ptoV, tipoCmp);**

Donde:  
- ptoV: es el punto de venta del comprobante  
- tipoCmp: es el código del tipo de comprobante

Ejemplo:

$numero=$webservice->UltimoCmp("2", "6");

En la variable $numero, se encuentra el último número del comprobante con punto de venta ”2” y tipo de comprobante “6”, donde “6” es el código de Factura “B”.

Por ejemplo cuando estamos completando los campos para la generación de una factura “B” punto de venta “2”, cuando tenemos que indicar el número de factura a generar, se podrá poner:

$webservice->factura->numeroComprobante=$webservice->UltimoCmp("2", "6")+1;

De esta manera, estamos sumando 1(uno) al último comprobante generado para que continúe con la numeración correlativa de dicha factura y no ocasionar que la AFIP la rechace por no ser correlativa.

**Método para obtener información de un comprobante generado:**

Con este método podemos obtener toda la información relacionada a un comprobante o simplemente saber si el comprobante existe. El método es:

**ConsultarComprobante(ptovta, tipoCmp, numero);**

Donde:  
- ptovta: es el punto de venta del comprobante  
- tipoCmp: es el código del tipo de comprobante  
- numero: es el número del comprobante a consultar

Ejemplo: Consultamos una Factura “A” Nº 00002-00000026

**$cmp**=$webservice-> ConsultarComprobante("2", "1", "26")

***Si el comprobante EXISTE:***  
En la variable **$cmp**, se encuetra la respuesta en XML que envía la afip del comprobante consultado.  
Y en el array:

- comprobante["CAE"] Contiene el CAE del comprobante consultado  
- comprobante["CAEVto"] Contiene el vencimiento del CAE del comprobante consultado

***Si el comprobante NO EXISTE:***  
La variable $cmp va a ser nula, por lo cual nos indica que el comprobante consultado no existe.  
Podemos capturar el tipo de error consultando el **array “errores”:**

*Descripción del array:*  
errores["ErrorCodigo"] – Devuelve el código de error  
errores["ErrorMensaje"] – Devuelve el mensaje del error

Nota:  
Este método es muy importante, no solo para consutar comprobantes generados, sino también por si en el proceso de autorización cuando se genera un comprobante, hay algún error de conexión. En este caso, no sabemos si se generó o no el comprobate.   
Con este metodo, podemos consultar si dicho comprobante se generó, en el caso que sí, se captura el CAE y su vencimiento.