



## Contenido

1	Control de cambios .....	2
2	Prefacio .....	2
2.1	Acerca de esta guía .....	2
2.2	Audiencia.....	2
2.3	Feedback para esta documentación .....	2
3	Transacción Oneclick.....	3
3.1	Descripción de la Transacción de Autorización Completa .....	3
3.2	Inscripción en Oneclick.....	4
3.3	Eliminar registro de inscripción de usuario.....	7
3.4	Transacción de Pago Oneclick .....	8
3.5	Reversa de pagos autorizados.....	9
3.6	Definición detallada de métodos del Servicio Web .....	10
4	Anexo: Generación de certificados digitales.....	12
5	Anexo: Ejemplos de integración con API SOAP Webpay.....	13

# 1 Control de cambios

Fecha	Versión	Descripción del cambio
17-01-2013	1.0	<ul style="list-style-type: none"><li>• Versión inicial nuevo formato con ejemplo PHP.</li></ul>
01-02-2013	1.1	<ul style="list-style-type: none"><li>• Se entregan observaciones adicionales en algunos puntos y se incorpora anexo de generación de llaves.</li></ul>
09-07-2013	1.2	<ul style="list-style-type: none"><li>• Se elimina método reverse obsoleto</li></ul>
04-09-2014	1.3	<ul style="list-style-type: none"><li>• Se actualizan valores de Excepción para restricciones One Click</li></ul>

## 2 Prefacio

### 2.1 Acerca de esta guía

Esta guía describe los aspectos técnicos que deben ser considerados en la integración con Webpay utilizando API SOAP, describe el servicio Web para Transacción Oneclick, sus operaciones y cómo estas deben ser utilizadas en la inscripción y en un flujo de pago. Al final de este documento se incluye un ejemplo en lenguaje PHP el cual puede ser tomado como guía.

### 2.2 Audiencia

Esta guía está dirigida a implementadores que realizan la integración de Webpay en comercios utilizando la API SOAP para soportar en estos el pago con tarjetas bancarias.

Se recomienda que quién realice la integración posea conocimiento técnico de al menos en los siguientes temas:

- Servicios Web
- WS-Security
- Firma digital, generación y validación.

### 2.3 Feedback para esta documentación

Ayúdanos a mejorar esta información enviándonos comentarios a [soporte@webpay.cl](mailto:soporte@webpay.cl)

## 3 Transacción Oneclick

### 3.1 Descripción de la Transacción de Autorización Completa

La modalidad de pago Oneclick permite al tarjetahabiente realizar pagos en el comercio sin la necesidad de ingresar cada vez información de la tarjeta de crédito al momento de realizar la compra. El modelo de pago contempla un proceso previo de inscripción o enrolamiento del tarjetahabiente, a través del comercio, que desee utilizar el servicio. Este tipo de pago facilita la venta, disminuye el tiempo de la transacción y reduce los riesgos de ingreso erróneo de los datos del medio de pago.

El proceso de integración con Webpay Oneclick consiste en desarrollar por parte del comercio las llamadas a los servicios web dispuestos por Transbank para la inscripción de los tarjetahabientes, así como para la realización de los pagos.

Resumen de los métodos del servicio Web

Método	Descripción general
<b>initInscription</b>	<p>Permite realizar la inscripción del tarjetahabiente e información de su tarjeta de crédito. Retorna como respuesta un token que representa la transacción de inscripción y una URL (UrlWebpay), que corresponde a la URL de inscripción de One Click.</p> <p>Una vez que se llama a este servicio Web, el usuario debe ser redireccionado vía POST a urlWebpay con parámetro TBK_TOKEN igual al token obtenido.</p>
<b>finishInscription</b>	<p>Permite finalizar el proceso de inscripción del tarjetahabiente en Oneclick. Entre otras cosas, retorna el identificador del usuario en Oneclick, el cual será utilizado para realizar las transacciones de pago.</p> <p>Una vez terminado el flujo de inscripción en Transbank el usuario es enviado a la URL de fin de inscripción que definió el comercio. En ese instante el comercio debe llamar a finishInscription.</p>
<b>authorize</b>	<p>Permite realizar transacciones de pago. Retorna el resultado de la autorización. Este método que debe ser ejecutado, cada vez que el usuario selecciona pagar con Oneclick.</p>
<b>codeReverseOneClick</b>	<p>Permite reversar una transacción de venta autorizada con anterioridad. Este método retorna como respuesta un identificador único de la transacción de reversa.</p>

## 3.2 Inscripción en Oneclick

La inscripción es el proceso en el cual el tarjetahabiente registra los datos de su tarjeta en Webpay Oneclick para usarlo en compras futuras. Estos datos son almacenados de forma segura en Transbank, y nunca son conocidos por el comercio.

Este proceso debe ser iniciado por la tienda del comercio y es requisito que el cliente esté autenticado en la página del comercio antes de iniciar la inscripción.

Proceso:

- El cliente se conecta y autentica en la página del comercio, mediante su nombre de usuario y clave.
- El cliente selecciona la opción de inscripción, la cual debe estar explicada en la página del comercio.
- El comercio consume un servicio web publicado por Transbank, donde entrega los datos del cliente y la URL de término; obtiene un token y URL de Webpay.
- El comercio envía el browser del cliente a la URL obtenida y pasa por parámetro el token (método POST).
- Webpay presenta el formulario de inscripción, este es similar al formulario de pago actual de Webpay Plus, para que el cliente ingrese los datos de su tarjeta.
- El cliente será autenticado por su banco emisor, de forma similar al flujo normal de pago. En este punto se realiza un transacción de \$1 peso, la cual no se captura (no se verá reflejada en su estado de cuenta).
- Finalizada la inscripción, Webpay envía el browser del cliente a la URL entregada por el comercio, pasando por parámetro el token.
- El comercio debe consumir otro servicio web de Transbank, con el token, para obtener el resultado de la inscripción y el identificador de usuario, que debe utilizar en el futuro para realizar los pagos.
- El comercio presenta al cliente el resultado de la inscripción.

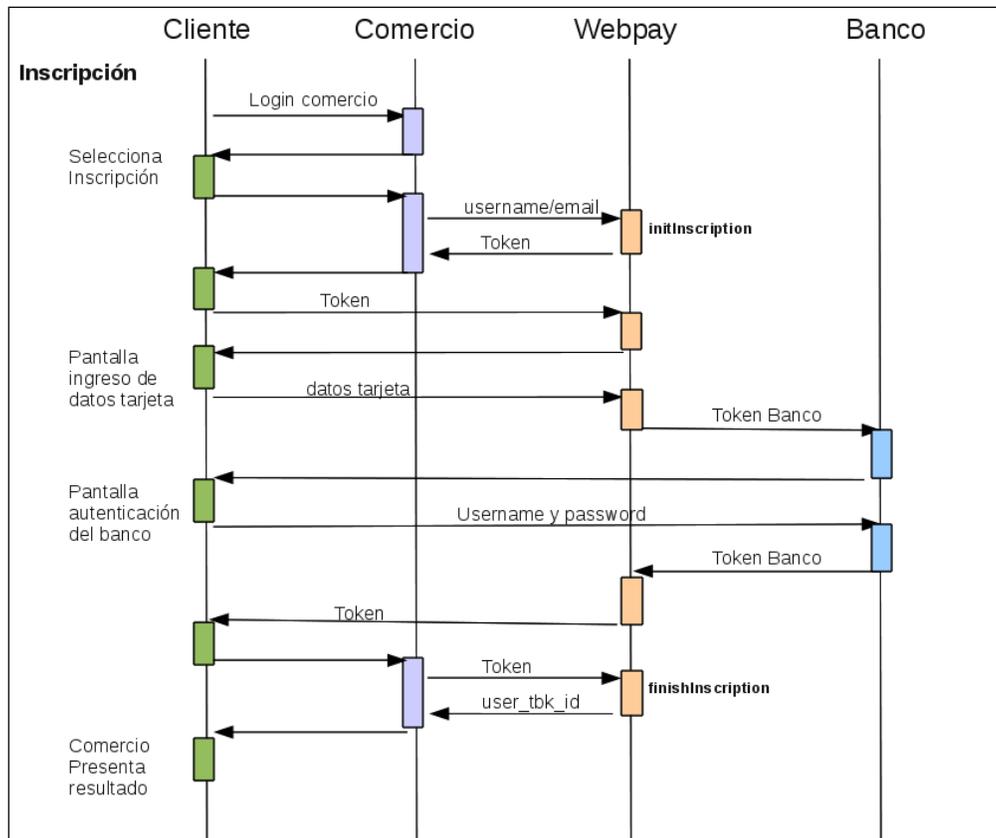


Diagrama de flujo proceso de inscripción

Método Web Service	
<b>initInscription</b>	: Permite gatillar el inicio del proceso de inscripción
<b>Recibe</b>	: {username}, {email}, {URL de fin de la inscripción}
	Username : String (255) Email : String (255) URL de fin de inscripción : String (255)
<b>Retorna</b>	: {token}, {urlWebpay}.
	Token : String (64) URL Webpay : String (255)
<b>Notas</b>	: Una vez que se llama a este webservice el usuario debe ser redireccionado vía POST a urlWebpay con parámetro TBK_TOKEN igual al token.

Método Web Service	
<b>finishInscription</b>	: finaliza el proceso de inscripción
<i>Recibe</i>	: {token}
	Token : String (64)
<i>Retorna</i>	: {id de usuario generado por Transbank}, {marca de tarjeta}, {últimos 4 dígitos}, {código de respuesta}, {código de autorización}
	Id de usuario : String (255)
	Marca de tarjeta : String (20)
<i>Notas</i>	: Una vez terminado el flujo de inscripción en Transbank el usuario es enviado a la URL de fin de inscripción que definió el comercio. En ese instante el comercio debe llamar a finishInscription.

### 3.3 Eliminar registro de inscripción de usuario

En el caso que el comercio requiera eliminar la inscripción de un usuario en Webpay OneClick ya sea por la eliminación de un cliente en su sistema o por la solicitud de este para no operar con esta forma de pago, el comercio deberá consumir un servicio web de Transbank con el identificador de usuario entregado en la inscripción.

Método Web Service	
<b>removeUser</b>	: Permite eliminar una inscripción de usuario en Transbank
<i>Recibe</i>	: {username}, {id usuario Transbank}
	Username : String (255)
	Id usuario Transbank : String (255)
<i>Retorna</i>	: { booleano que indica si la inscripción fue dada de baja correctamente)
	Return : boolean (true/false)

### 3.4 Transacción de Pago Oneclick

El pago es el proceso donde el comercio solicita el cargo de una compra a la tarjeta de crédito de un usuario inscrito anteriormente, usando el identificador entregado por Transbank al momento de la inscripción.

Los pagos en esta modalidad no requieren la intervención del usuario.

El monto del pago debe estar dentro de los límites establecidos para este tipo de transacciones, el proceso interno es similar a un cargo normal de Webpay.

Proceso:

- El cliente se conecta y autentica en la página o aplicación del comercio mediante su nombre de usuario y clave.
- El cliente selecciona la opción de pagar con Webpay Oneclick.
- El comercio usa el servicio web de pago, publicado por Transbank, entregando el identificador de usuario (que se obtuvo en la inscripción), el monto del pago y la orden de compra. Obtiene la respuesta con el código de autorización.
- El comercio presenta el resultado del pago al cliente.

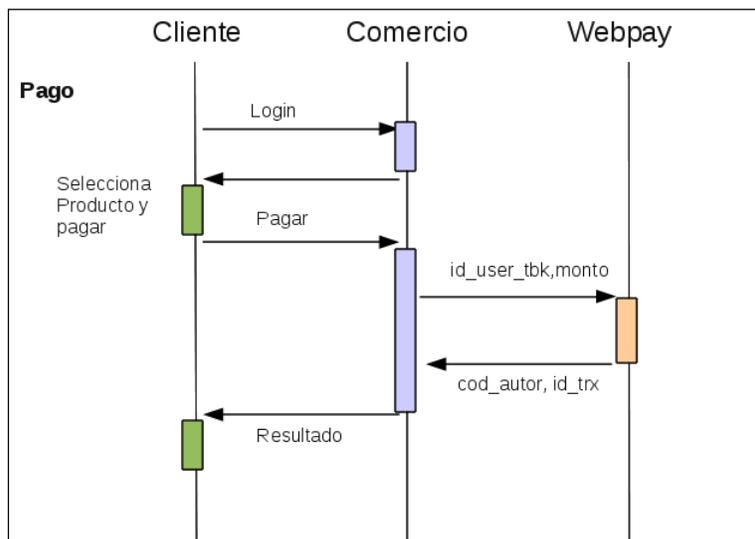


Diagrama de flujo proceso de pago

Método Web Service	
<b>authorize</b>	: Permite autorizar un pago.
<i>Recibe</i>	: {username}, {id usuario Transbank}, {monto}, {orden de compra}.
Username	: String (255)
Id usuario Transbank	: String (255)
Monto	: Number (19,2)
Orden de compra	: String (255)

<i>Retorna</i>	: {código de respuesta}, {código de autorización}, }, {marca de tarjeta}, {últimos 4 dígitos}, {id de la transacción}
	Codigo de respuesta : Number (10,0) Codigo de autorización : String (20) Marca de tarjeta : String (20) Ultimos 4 digitos : String (4)
<i>Notas</i>	: Método que debe ser ejecutado, cada vez que el usuario selecciona pagar con Oneclick

### 3.5 Reversa de pagos autorizados

Este proceso permite revertir una venta cuando esta no pudo concretarse, dentro del mismo día contable, con la finalidad de anular un cargo realizado al cliente.

El comercio, en caso de requerir revertir un pago, debe consumir un servicio web publicado por Transbank con el identificador del pago entregado en la respuesta de la autorización de la transacción.

Método Web Service	
<b>codeReverseOneClick</b>	: Permite revertir una venta y se obtiene el resultado de la reversa y un código de reversa en caso que esta sea exitosa
<i>Recibe</i>	: {orden de compra}
	Orden de compra : Long
<i>Retorna</i>	: {reverseCode} {reversed}
	reverseCode : Long
	reversed : boolean (true/false)
<i>Nota</i>	: El código regresado por este método, es un identificador único de la transacción de reversa.

### 3.6 Definición detallada de métodos del Servicio Web

initInscription	
INPUT	
<b>email</b>	: Dirección de correo electrónico registrada por el comercio, si no la tiene registrada es necesario solicitarla antes.
<b>username</b>	: Nombre de usuario o cliente en el sistema del comercio.
<b>responseURL</b>	: URL de retorno, donde será enviado el cliente, finalizado el proceso de inscripción.
OUTPUT	
<b>Token</b>	: Identificador, único, del proceso de inscripción, debe ser enviado por parámetro (TBK_TOKEN) a la URL de Webpay.
<b>urlWebpay</b>	: URL del sistema Webpay, para iniciar la inscripción.

finishInscription	
INPUT	
<b>token</b>	: Identificador del proceso de inscripción, entregado por Webpay en el método initInscription.
OUTPUT	
<b>responseCode</b>	: Código de retorno del proceso de inscripción, donde 0 (cero) es aprobado.
<b>authCode</b>	: Código que identifica la autorización de la inscripción.
<b>creditCardType</b>	: Indica el tipo de tarjeta que fue inscrita por el cliente ( Visa, AmericanExpress, MasterCard, Diners, Magna)
<b>last4CardDigits</b>	: Los últimos 4 dígitos de la tarjeta ingresada por el cliente en la inscripción.
<b>tbkUser</b>	: Identificador único de la inscripción del cliente, este debe ser usado para realizar pagos, o borrar la inscripción.

Authorize	
INPUT	
<b>amount</b>	: Monto del pago en pesos.
<b>tbkUser</b>	: Identificador único de la inscripción del cliente.
<b>username</b>	: Nombre de usuario, del cliente, en el sistema del comercio.
<b>buyOrder</b>	: Identificador único de la compra generado por el comercio. Debe ser timestamp [yyyymmddhhMMss] + un correlativo de tres dígitos. Ej: Para la tercera transacción realizada el día 15 de julio de 2011 a las 11:55:50 la orden de compra sería: 20110715115550003.
OUTPUT	
<b>responseCode</b>	: Código de retorno del proceso de pago, donde: 0 (cero) es aprobado. -1 Rechazo -2 Rechazo -3 Rechazo -4 Rechazo -5 Rechazo -6 Rechazo -7 Rechazo -8 Rechazo

	-97 limites Oneclick, máximo monto diario de pago excedido -98 limites Oneclick, máximo monto de pago excedido -99 limites Oneclick, máxima cantidad de pagos diarios excedido
<b>authCode</b>	: Código de autorización de la transacción de pago.
<b>last4CardDigits</b>	: Los últimos 4 dígitos de la tarjeta ingresada por el cliente en la inscripción.
<b>creditCardType</b>	: Indica el tipo de tarjeta que fue inscrita por el cliente ( Visa, AmericanExpress, MasterCard, Diners, Magna)
<b>transactionId</b>	: Identificador único de la transacción de pago, se utiliza para la reversa, si fuera necesario.

removeUser	
INPUT	
<b>tbkUser</b>	: Identificador único de la inscripción del cliente.
<b>username</b>	: Nombre de usuario, del cliente, en el sistema del comercio.
OUTPUT	
booleano	: Retorno verdadero, si fue posible eliminar la inscripción. Falso de lo contrario.

## 4 Anexo: Generación de certificados digitales

Para las siguientes instrucciones se utiliza un comercio de ejemplo, el cual su código es 597029124456. Favor reemplace el código de comercio 597029124456 por el asignado por Transbank.

### 1. Crear llave privada

```
openssl genrsa -out 597029124456.key 2048
```

El comercio es responsable de tomar los resguardos necesarios con la llave privada.

### 2. Crear requerimiento de certificado

```
openssl req -new -key 597029124456.key -out 597029124456.csr
```

```
Country Name (2 letter code) []:CL
State or Province Name (full name) []:
Locality Name (eg, city) []:SANTIAGO
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:597029124456
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

### 3. Crear certificado autofirmado

```
openssl x509 -req -days 1460 -in 597029124456.csr -signkey
597029124456.key -out 597029124456.crt
```

Este certificado tiene una validez de 1460 días (4 años) el cual debe ser renovado antes que expire.

El comercio debe enviar a Transbank el certificado publico (597029124456.crt) y quedarse con su llave privada con la cual deberá transaccionar.

## 5 Anexo: Ejemplos de integración con API SOAP Webpay

Los siguientes ejemplos tienen por objetivo exponer una forma factible de integración con API SOAP Webpay para resolver los siguientes puntos asociados a la integración:

1. Generación de cliente o herramienta para consumir los servicios Web, lo cual permite abstraerse de la complejidad de mensajería SOAP asociada a los Webservice y hacer uso de las operaciones del servicio.
2. Firma del mensaje y validación de firma en la respuesta, existen frameworks y herramientas asociadas a cada lenguaje de programación que implementan el estándar WS Security, lo que se requiere es utilizar una de estas, configurarla y que realice el proceso de firma digital del mensaje.

El siguiente ejemplo está basado en PHP versión 5, sobre el cual se utilizaron las siguientes bibliotecas de software para realizar la invocación de los servicios web de Webpay bajo el estándar WSS:

- **Biblioteca de seguridad:** archivo compuesto de tres clases que integran librerías nativas PHP de validación y verificación. Estas clases nos permitirán generar la seguridad suficiente a través de métodos de encriptación y desencriptación.
- **WSSE-PHP:** integra las librerías de seguridad XML y genera un documento XML-SOAP seguro. Depende de las librerías de seguridad XML.
- **SOAP-VALIDATION:** clase encargada de la validación de mensajes SOAP seguros de respuesta. Verifica la autenticidad e integridad del mensaje. Depende de WSSE-PHP.

Los fuentes pueden ser descargados desde <https://github.com/OrangePeople/php-wss-validation>

Pasos a seguir:

### 1. Generación de cliente del Servicio Web:

Antes de la integración se debe tener instalado y funcionando un servidor HTTP Apache y configurar el directorio para generar una salida a través del navegador web. Si se quiere optar por una alternativa más sencilla, Apache provee un directorio por omisión, que varía según el sistema operativo. Generalmente en plataformas Linux es “/var/www” y en Windows “C:\<directorio hacia Apache>/htdocs”.

A continuación, para generar las clases necesarias que conectan a los servicios Web, se puede utilizar la herramienta Easy WSDL2PHP. La documentación necesaria e información de descarga se encuentra en <http://sourceforge.net/projects/easywsdl2php/>.

Una vez descargados los fuentes, se deben copiar en el directorio de apache que posee la salida por navegador.

Se hace la llamada por navegador del archivo **wsdl2php.php** y se obtiene la siguiente pantalla:

## Easy WSDL2PHP Generator

ej. <http://soap.amazon.com/schemas2/AmazonWebServices.wsdl>

Url:

Class Name:

Se escribe la URL del **archivo wsdl** al se quiere conectar, un **nombre de clase** y luego se presiona el botón **Generate Code**. Luego de esto se muestra una pantalla como la siguiente:

## Easy WSDL2PHP Generator

ej. <http://soap.amazon.com/schemas2/AmazonWebServices.wsdl>

Url:

Class Name:

```
<?php
class removeUser{
var $arg0;//oneClickRemoveUserInput
}
class oneClickRemoveUserInput{
var $tbkUser;//string
var $username;//string
}
class baseBean{
}
class removeUserResponse{
var $return;//boolean
}
class initInscription{
var $arg0;//oneClickInscriptionInput
}
class oneClickInscriptionInput{
var $email;//string
var $responseURL;//string
var $username;//string
}
```

Una vez que se obtiene el resultado mostrado en la imagen se copia el código PHP generado y se guarda en un archivo, el cual representará el stub del servicio Web. Una vez realizado este proceso ya se tienen las clases necesarias para poder integrarse con los servicios web de Webpay.

**2. Crear una clase que extienda de SoapClient** (*SoapClient es la clase nativa que provee PHP para utilización de servicios Web*)(En el ejemplo se denominará MySoap)

```
//Notar que se incluyen dos archivos que se proveen en La Librería de encriptación
require_once('xmlseclibs.php');
require_once('soap-wsse.php');
```

```
class MySoap extends SoapClient {

    function __doRequest($request, $Location, $saction, $version) {
        $doc = new DOMDocument('1.0');
        $doc->LoadXML($request);
        $objWSSE = new WSSESoap($doc);
        $objKey = new XMLSecurityKey(XMLSecurityKey::RSA_SHA1,array('type' =>
            'private'));
        $objKey->LoadKey(PRIVATE_KEY, TRUE);
        $options = array("insertBefore" => TRUE);
        $objWSSE->signSoapDoc($objKey, $options);
        $objWSSE->addIssuerSerial(CERT_FILE);
        $objKey = new XMLSecurityKey(XMLSecurityKey::AES256_CBC);
        $objKey->generateSessionKey();
        $retVal = parent::__doRequest($objWSSE->saveXML(), $Location, $saction,
            $version);
            $doc = new DOMDocument();
            $doc->LoadXML($retVal);
            return $doc->saveXML();
        }
    }
}
```

Las constantes **PRIVATE\_KEY** Y **CERT\_FILE** son las rutas de la llave privada y certificado del comercio, respectivamente. Es importante que siempre se trabaje con estos dos elementos.

### 3. Incluir la clase generada en el paso anterior en el archivo principal de los servicios.

Se debe incluir con la sentencia `require_once` la clase generada en el paso anterior.

Ejemplo:

```
require_once("mysoap.php");
```

### 4. Editar el archivo stub creado en el paso 1

Se debe editar el método `__construct` del stub tal como se muestra en el ejemplo:

**Donde dice:**

```
$this->soapClient = new SoapClient($url, array("classmap" => self::$classmap, "trace" => true, "exceptions" => true));
```

**Debe quedar: (utilizando el nombre de clase del paso 2)**

```
$this->soapClient = new MySoap($url, array("classmap" => self::$classmap, "trace" => true, "exceptions" => true));
```

### 5. Invocación de operaciones del servicio web de Webpay.

#### Observaciones

1. Para todos los ejemplos se debe hacer referencia a los archivos de la librería descargada

```
require_once('soap-wsse.php');
require_once('soap-validation.php');
require_once('<archivo que contiene la clase stub creado en el paso 1>');
```

2. El objeto **OneClickWS** representa la clase stub generada.

3. El objeto **oneClickInscriptionResponse** contiene la respuesta a la operación realizada.

4. El método **getValidationResult** indica si el mensaje está firmado por Transbank. Esto debe validarse siempre.

5. La constante **SERVER\_CERT** representa el path del certificado entregado por Transbank para realizar la integración.

### Operación initInscription

```
define("SERVER_CERT", dirname(__FILE__) . "/certs/oneclick.pem");
$oneClickService = new OneClickWS();
$oneClickInscriptionInput = new oneClickInscriptionInput();

$oneClickInscriptionInput->username = $_POST['username'];
$oneClickInscriptionInput->email = $_POST['email'];
$oneClickInscriptionInput->responseURL = "<url comerico resultado inscripcion>";

$oneClickInscriptionResponse = $oneClickService->initInscription(array("arg0" =>
$oneClickInscriptionInput));
$xmlResponse = $oneClickService->soapClient->__getLastResponse();
$soapValidation = new SoapValidation($xmlResponse, SERVER_CERT);

$soapValidation->getValidationResult(); //Esto valida si el mensaje está firmado
por Transbank

$oneClickInscriptionOutput = $oneClickInscriptionResponse->return; //Esto obtiene
el resultado de la operación

$tokenOneClick = $oneClickInscriptionOutput->token; //Token de resultado
$inscriptionURL = $oneClickInscriptionOutput->urlWebpay; //URL para realizar el
post
```

### Operación finishInscription

```
define("SERVER_CERT", dirname(__FILE__) . "/certs/oneclick.pem");

$oneClickService = new OneClickWS();
$oneClickFinishInscriptionInput = new oneClickFinishInscriptionInput();
$oneClickFinishInscriptionInput->token = $tokenOneClick; // $tokenOneClick es el
token de resultado obtenido en el metodo initInscription.

$oneClickFinishInscriptionResponse =
    $oneClickService->finishInscription(array(
        "arg0" => $oneClickFinishInscriptionInput));

$xmlResponse = $oneClickService->soapClient->__getLastResponse();
$soapValidation = new SoapValidation($xmlResponse, SERVER_CERT);
//Si la firma es válida
$oneClickFinishInscriptionOutput = $oneClickFinishInscriptionResponse->return;
//Datos de resultado de la inscripción OneClick
$responseCode = $oneClickFinishInscriptionOutput->responseCode;
$authCode = $oneClickFinishInscriptionOutput->authCode;
$creditCardType = $oneClickFinishInscriptionOutput->creditCardType;
$last4CardDigits = $oneClickFinishInscriptionOutput->last4CardDigits;
$tbkUser = $oneClickFinishInscriptionOutput->tbkUser;
```

### Operación authorize

```
define("SERVER_CERT", dirname(__FILE__) . "/certs/oneClick.pem");

$oneClickService = new OneClickWS();
$oneClickInscriptionInput = new oneClickInscriptionInput();
$oneClickPayInput = new oneClickPayInput();

$oneClickPayInput->amount = <monto de pago>;
$oneClickPayInput->buyOrder = <orden de compra>;
$oneClickPayInput->tbkUser = <identificador de usuario entregado
                             en el servicio finishInscription>;

$oneClickPayInput->username = <identificador de usuario del comercio>;

$oneClickauthorizeResponse = $oneClickService->authorize(array
    ("arg0" => $oneClickPayInput));

$xmlResponse = $oneClickService->soapClient->__getLastResponse();
$soapValidation = new SoapValidation($xmlResponse, SERVER_CERT);
$oneClickPayOutput = $oneClickauthorizeResponse->return;

//Resultado de La autorización
$authorizationCode = $oneClickPayOutput->authorizationCode;
$creditCardType = $oneClickPayOutput->creditCardType;
$last4CardDigits = $oneClickPayOutput->last4CardDigits;
$responseCode = $oneClickPayOutput->responseCode;
$transactionId = $oneClickPayOutput->transactionId;
```

### **Operación codeReverseOneClickResponse (Esta operación reversa y retorna código de reversa)**

```
define("SERVER_CERT", dirname(__FILE__) . "/certs/oneclick.pem");

$oneClickService = new OneClickWS();
$oneClickReverseInput = new oneClickReverseInput();

$buyOrder = <orden de compra de la transacción>;
$oneClickReverseInput->buyorder= $buyOrder;
$codeReverseOneClickResponse = $oneClickService->codeReverseOneClick(array("arg0"
=> $oneClickReverseInput));
$oneClickReverseOutput = $codeReverseOneClickResponse->return;
$xmlResponse = $oneClickService->soapClient->__getLastResponse();
$soapValidation = new SoapValidation($xmlResponse, SERVER_CERT);
//Si la firma es válida
$reversed = $oneClickReverseOutput->reversed; //Indica si la reversa se ejecutó
$reverseCode = $oneClickReverseOutput->reverseCode; //Entrega Código de reversa
```

### **Operación removeUser**

```
define("SERVER_CERT", dirname(__FILE__) . "/certs/oneclick.pem");

$oneClickService = new OneClickWS();
$oneClickRemoveUserInput = new oneClickRemoveUserInput();

$tbkUser = <identificador de usuario entregado en el servicio finishInscription>;
$commerceUser = <identificador de usuario del comercio>;

$oneClickRemoveUserInput->tbkUser = $tbkUser;
$oneClickRemoveUserInput->username = $commerceUser;

$removeUserResponse = $oneClickService->removeUser(array
("arg0" => $oneClickRemoveUserInput));

$xmlResponse = $oneClickService->soapClient->__getLastResponse();
$soapValidation = new SoapValidation($xmlResponse, SERVER_CERT);
//Si la firma es válida
$removeUserResponse->return;
// Valor booleano que indica si el usuario fue removido.
```