Método de conexión a Place to Pay

Webservice PSE - ACH Colombia

NIT: 900.299.228-0 • Calle 53 N° 45- 112 Of. 1901 • Tel.+57(4) 444 23 10 • www.placetopay.com
Place to Pay, una marca de EGM Ingeniería sin Fronteras

Tabla de contenidos

Información del documento

Nombre del documento	Método de conexión a PlacetoPay Webservice PSE - ACH Colombia
Versión	2.2.1
Autor	EGM Ingeniería sin fronteras

Control de versiones

Versión	Fecha	Adiciones / Modificaciones	Preparado por
2.0.0	2013-01-16	Generación del nuevo documento de especificación del servicio con mecanismo de autenticación adicional.	Ingeniería
2.1.0	2013-04-22	Modificadas las estructuras TransactionInformation y PSETransactionResponse para adicionar el campo sessionID	Ingeniería
2.1.1	2013-05-21	Corregido error en la documentación que mostraba un campo doble en la estructura TransactionInformation.	Ingeniería
2.1.2	2014-07-27	Especificación de los tamaños de las cadenas de caracteres	Ingeniería
2.1.3	2015-08-20	Cambiado el código del documento	Ingeniería
2.2.1	2016-04-05	Adición de método para crear transacciones multicrédito y sus correspondientes estructuras	Ingeniería

Introducción

Este Webservice y su arquitectura permiten que cualquier aplicativo sin importar el lenguaje de desarrollo, puedan beneficiarse de un conjunto de reglas de negocio y servicios, debido al cumplimiento de estándares SOAP, WSDL y XML.

Para hacer uso del Webservice, deberá contarse con un lenguaje de programación que pueda consumirlo, algunos lenguajes tienen mejor soporte que otros desde el punto de vista de facilidad de acceso; sin embargo, si no se cuenta con un lenguaje que encapsule la comunicación, siempre se puede hacer incluso a nivel de sockets usando el protocolo HTTP conjuntamente con la especificación SOAP.

Para aquellos lenguajes que tienen un buen soporte lo más aconsejable es ir a través del WSDL para el Webservice.

¿Cómo funciona la implementación?

1. Se muestra como opción de pago PSE (Débitos a cuentas de ahorro y corriente en Colombia).
2. Una vez el usuario la selecciona, se presenta la lista de bancos (debe estar cacheada localmente y su refrescamiento debe ser 1 vez por día) y el tipo de banca a desplegar (personas o empresas).
3. Al tiempo que el cliente confirma la operación, debe consumirse el servicio para realizar la petición de la transacción.
4. Si la petición es exitosa se retorna la URL a la cual debe enviarse al cliente para que realice la operación en el banco. En caso contrario se retorna el motivo del rechazo de la transacción.
5. Almacenar los datos de la transacción retornados por el servicio (identificador de la transacción, autorizacion/cus, identificador sesión de placetopay).
6. Redirigir el navegador a la URL retornada en caso de exito.
7. Una vez la transacción retorna (retorna a la URL especificada en el consumo del crear transacción), se debe preguntar por el estado de la transacción mediante el consumo un servicio.
8. Dependiendo de la respuesta del servicio, la transacción puede haber sido aprobada, rechazada o seguir pendiente de procesamiento.
9. Informar al usuario el estado de la transacción.

NIT: 900.299.228-0 • Calle 53 N° 45- 112 Of. 1901 • Tel.+57(4) 444 23 10 • www.placetopay.com
Place to Pay, una marca de EGM Ingeniería sin Fronteras

10. Si la transacción está pendiente, o si el cliente al abandonar el portal no ha regresado, se debe tener una sonda que pregunte por el estado de la transacción.

¿En qué casos es mejor usar este tipo de integración?

La integración directa al medio de pago en general NO es recomendada, pues dificulta otros servicios que pueden estar implementados en la interfaz de redirección de Place to Pay, al tiempo que hace que cada nuevo medio de pago liberado requiera un nuevo desarrollo por parte del comercio. Así que solo tener en cuenta esta integración si:

- Se requiere un completo flujo de transacción no disponible por redirección.
- Se requiere usar el servicio a modo de marca blanca.

¿Qué obligaciones tengo al usar esta integración?

Tenga en cuenta que al usar este tipo de integración, igual requiere de certificación y los tiempos de implementación pueden ser sustancialmente más altos que cuando utiliza el mecanismo de redirección.

Así mismo, se recomienda contar con un certificado digital SSL para su sitio.

Datos para la operación

A continuación se describen las interfaces básicas con las cuales debe contar su desarrollo para poder pasar exitosamente la revisión del proceso.

Captura de la información del pagador y del comprador

Al consumir el servicio de creación de la transacción, su aplicativo deberá proveer información de quién es el pagador y quién el comprador. Si corresponden a la misma persona, simplemente deberá especificar la información del pagador al solicitar la creación de la transacción.

Para poder proveer esta información al servicio, su aplicativo deberá capturarlo directamente en el proceso o de alguna fuente de información previamente habilitada.

Despliegue de la información de bancos disponibles

Una vez el cliente ha determinado que desea pagar con débito a una cuenta corriente o de ahorros, deberá presentarle la lista de los bancos que en el momento están disponibles para la transacción. Esta lista de bancos se obtiene mediante el consumo del método getBankList el cual debe ser consumido una vez por día, lo que exige que se disponga de un mecanismo de caché que permita recuperar la lista de este si ya ha sido consumido el servicio en el día.

Adicional a darle a escoger con qué banco realizar la transacción, el usuario también debe indicar qué tipo de interfaz del banco desea desplegar, es decir, si la de personas o la de empresas.

Si por algún motivo la lista de bancos no pudo ser obtenida, deberá mostrarse un mensaje de "No se pudo obtener la lista de Entidades Financieras, por favor intente más tarde" y, consumir nuevamente el servicio de lista de bancos para intentar obtener dicha lista y poderla almacenar en el caché.

Tenga en cuenta que en operaciones de Multicrédito se debe utilizar el código de servicio para multicrédito y que siempre deberá pasar todas las cuentas dependientes (entidad, servicio) para los créditos así algunas de ellas vayan con valores en cero. Es requerido que la lista de créditos corresponda con todos los subcódigos dependientes.

Confirmación y redirección al banco

Una vez se poseen los datos del pagador y comprador, así como la información de qué banco y qué tipo de interfaz debe mostrar el banco deberá entonces consumir el servicio de createTransaction para obtener la URL a la cual deberá redirigir al cliente para finalizar la transacción.

Debe tener en cuenta que en los datos solicitados para crear la transacción se requiere la dirección IP desde la cual el cliente está realizando la transacción, así como la información del agente de navegación usado. Si lo desea, también puede proveer datos adicionales con la transacción, los cuales le permitirán tener esta información disponible en la consola de consulta de transacciones de Place to Pay.

Si la respuesta del servicio es **SUCCESS**, entonces deberá almacenar la información retornada en su base de datos, de vital importancia el **transactionID** pues es con este identificador que podrá consultar el estado final de la transacción.

NIT: 900.299.228-0 • Calle 53 N° 45- 112 Of. 1901 • Tel.+57(4) 444 23 10 • www.placetopay.com
Place to Pay, una marca de EGM Ingeniería sin Fronteras

Tenga en cuenta que una respuesta **SUCCESS** en este punto sólo implica que la transacción ha sido aprovisionada por el banco y está en espera que el cliente llegue a la interfaz del banco, se autentique y autorice la operación iniciada.

Al crear la transacción, esta puede fallar; algunos motivos incluyen montos por fuera del rango permitido, no disponibilidad del banco, problemas de configuración de la cuenta recaudadora, entre otros. Utilice la propiedad **responseReasonText** para obtener el mensaje de por qué falló la creación de la transacción. Algunos códigos no relacionados con el pagador pueden indicarle que genere una alerta sobre problemas con la disponibilidad del servicio, particularmente los relacionados con mala configuración.

Una vez ha almacenado la información en su base de datos y ha confirmado que es exitosa la petición, se redirigirá al cliente a la URL del banco. Tenga en cuenta que la redirección a la interfaz del banco no debe realizarse en un frame o cualquier otro mecanismo que oculte la URL en la cual el cliente ingresará sus datos de autenticación.

Reingreso del cliente

Una vez el cliente ha finalizado la transacción en la interfaz del banco, el banco redirige al cliente a la URL especificada al crear la transacción, en este punto de entrada a su aplicativo, usted deberá consumir el servicio getTransactionInformation que le permite determinar el estado de la transacción. Solo si la transacción tiene como estado **OK** es porque la transacción fue autorizada, si por el contrario obtiene un estado **PENDING** es porque aún no tiene respuesta final del banco acerca de la transacción.

Ya sea que el cliente retorne al punto de reingreso y que la transacción esté pendiente, o que haya abandonado la operación y haya roto el flujo normal, se deberá consumir el servicio getTransactionInformation para todas las operaciones que desde que fue invocado el createTransaction llevan más de 7 minutos sin un estado final de operación.

Descripción de la interfaz del Webservice

A continuación se describen las operaciones (métodos) que el webservice brinda. Las operaciones descritas a continuación son tipo petición – respuesta.

A continuación se describen cada una de las operaciones expuestas por el Webservice, se muestran los parámetros de entrada a cada operación y vínculos a las estructuras de datos usadas.

getBankList

Obtiene la lista de bancos disponibles para el establecimiento de comercio en el sistema PSE de ACH Colombia.

Tenga en cuenta que la respuesta de este servicio debe ser cacheada, el servicio debe ser consumido una vez por día.

Parámetros

Nombre	Tipo	Descripción
auth	Authentication	datos de autenticación

Retorno

Bank[]. Un arreglo con la lista de bancos habilitados.

createTransaction

Solicita la creación de una transacción. En los datos de la solicitud se especifica quién es el pagador, el comprador y el despacho. Así mismo para cuál de los bancos habilitados se hace la petición y a que URL de retorno debe el banco redirigir al cuenta habiente.

Parámetros

Nombre	Tipo	Descripción
auth	Authentication	datos de autenticación

transaction	PSETransactionRequest	datos de la solicitud

Retorno

PSETransactionResponse. Respuesta de la creación de la transacción, tenga en cuenta que la URL del banco se entrega solo si la propiedad returnCode es **SUCCESS**.

createTransactionMultiCredit

Solicita la creación de una transacción con dispersión de fondos. En los datos de la solicitud se especifica quién es el pagador, el comprador y el despacho. Así mismo para cuál de los bancos habilitados se hace la petición y a que URL de retorno debe el banco redirigir al cuenta habiente. Así como cada uno de los créditos a aplicar para cada uno de los servicios asociados. Tenga en cuenta que un servicio multicrédito tiene asociado unos servicios dependientes. Siempre deberá cobrar a todos los servicios dependientes así el valor para una de los créditos sea cero.

Parámetros

Nombre	Tipo	Descripción
auth	Authentication	datos de autenticación
transaction	PSETransactionMultiCredit Request	datos de la solicitud

Retorno

PSETransactionResponse. Respuesta de la creación de la transacción, tenga en cuenta que la URL del banco se entrega solo si la propiedad returnCode es **SUCCESS**.

getTransactionInformation

Obtiene la información de una transacción, debe ser consultado para cualquier operación previamente creada con el método createTransaction o createTransactionMultiCredit ya sea cuando regresa del banco o cuando al menos han transcurrido 7 minutos desde que el cliente fue redirigido a la interfaz del banco. Deberá consumirse en intervalos de al menos cada 12 minutos hasta que tenga un estado de transacción **transactionState** diferente a **PENDING**.

Parámetros

Nombre	Tipo	Descripción
auth	Authentication	datos de autenticación
transactionID	int	identificador único de la transacción en PlacetoPay, equivale al retornado en la creación de la transacción

Retorno

TransactionInformation. La información del estado de la transacción.

Tipos de datos o estructuras

En este apartado se describen cada una de las estructuras de datos usadas por los métodos del Webservice.

Attribute

Estructura para almacenar información extendida.

Propiedades

Nombre	Tipo	Descripción
name	string[30]	Código para referenciar el atributo
value	string[128]	Valor que asume el atributo

Authentication

Estructura para autenticarse con el webservice.

Tenga en cuenta que el **login** y el **tranKey** son dados por PlacetoPay para poder realizar transacciones en nombre de un sitio de recaudo.

La semilla usada en el consumo debe ser una fecha en formato ISO 8601 (AAAA-MM-DDTHH:NN:SSZZZ, ej: 2013-04-11T08:47:21-05:00), el sistema verificará que la petición no haya expirado, por ello es fundamental una sincronización del reloj del servidor. Más abajo en este documento hay un apartado de la obtención de la semilla en el formato solicitado y la generación del Hash.

El tranKey enviado en el consumo corresponde a un hash generado con SHA1 de la concatenación de la semilla y del tranKey originalmente enviado por PlacetoPay.

Propiedades

Nombre	Tipo	Descripción
login	string[32]	Identificador habilitado para el consumo del API, entregado por Place to Pay.
tranKey	string[40]	Llave transaccional para el consumo del API SHA1(seed + tranKey)
seed	string	Semilla usada para el consumo del API en el proceso del hash por SHA1 del tranKey, ISO 8601.
additional	Attribute[]	Datos adicionales a la estructura de autenticación

Person

Estructura para reflejar la información de una persona involucrada en una transacción

Propiedades

Nombre	Tipo	Descripción
document	string[12]	Número de identificación de la persona
documentType	string[3]	Tipo de documento de identificación de la persona [CC, CE, TI, PPN]. CC = Cédula de ciudanía colombiana CE = Cédula de extranjería TI = Tarjeta de identidad PPN = Pasaporte

		NIT = Número de identificación tributaria SSN = Social Security Number
firstName	string[60]	Nombres
lastName	string[60]	Apellidos
company	string[60]	Nombre de la compañía en la cual labora o representa
emailAddress	string[80]	Correo electrónico
address	string[100]	Dirección postal completa
city	string[50]	Nombre de la ciudad coincidente con la dirección
province	string[50]	Nombre de la provincia o departamento coincidente con la dirección
country	string[2]	Código internacional del país que aplica a la dirección física acorde a ISO 3166-1, mayúscula sostenida.
phone	string[30]	Número de telefonía fija
mobile	string[30]	Número de telefonía móvil o celular

Bank

Estructura para reflejar la información de una entidad bancaria

Propiedades

Nombre	Tipo	Descripción
bankCode	string[4]	Código de la entidad financiera
bankName	string[60]	Nombre de la entidad financiera

CreditConcept

Estructura que representa el concepto del crédito a favor de un tercero

Propiedades

Nombre	Tipo	Descripción
entityCode	string[12]	Código de la entidad del tercero para dispersión
serviceCode	string[12]	Código del servicio del tercero
amountValue	double	Valor total a recaudar a favor de la entidad
taxValue	double	Discriminación del impuesto aplicado a favor de la entidad
description	string[60]	Descripción el concepto cobrado

PSETransactionRequest

Estructura que representa una solicitud de transacción con débitos a cuenta PSE.

Propiedades

Nombre	Tipo	Descripción
bankCode	string[4]	Código de la entidad financiera con la cual realizar la transacción
bankInterface	string[1]	Tipo de interfaz del banco a desplegar [0 = PERSONAS, 1 = EMPRESAS]
returnURL	string[255]	URL de retorno especificada para la entidad financiera
reference	string[32]	Referencia única de pago
description	string[255]	Descripción del pago
language	string[2]	Idioma esperado para las transacciones acorde a ISO 631-1, mayúscula sostenida
currency	string[3]	Moneda a usar para el recaudo acorde a ISO 4217

totalAmount	double	Valor total a recaudar
taxAmount	double	Discriminación del impuesto aplicado
devolutionBase	double	Base de devolución para el impuesto
tipAmount	double	Propina u otros valores exentos de impuesto (tasa aeroportuaria) y que deben agregarse al valor total a pagar
payer	Person	Información del pagador
buyer	Person	Información del comprador
shipping	Person	Información del receptor
ipAddress	string[15]	Dirección IP desde la cual realiza la transacción el pagador
userAgent	string[255]	Agente de navegación utilizado por el pagador
additionalData	Attribute[]	Datos adicionales para ser almacenados con la transacción

PSETransactionMultiCreditRequest

Estructura que representa una solicitud de transacción con débitos a cuenta PSE.

Propiedades

Nombre	Tipo	Descripción
bankCode	string[4]	Código de la entidad financiera con la cual realizar la transacción
bankInterface	string[1]	Tipo de interfaz del banco a desplegar [0 = PERSONAS, 1 = EMPRESAS]
returnURL	string[255]	URL de retorno especificada para la entidad financiera
reference	string[32]	Referencia única de pago

description	string[255]	Descripción del pago
language	string[2]	Idioma esperado para las transacciones acorde a ISO 631-1, mayúscula sostenida
currency	string[3]	Moneda a usar para el recaudo acorde a ISO 4217
totalAmount	double	Valor total a recaudar
taxAmount	double	Discriminación del impuesto aplicado
devolutionBase	double	Base de devolución para el impuesto
tipAmount	double	Propina u otros valores exentos de impuesto (tasa aeroportuaria) y que deben agregarse al valor total a pagar
payer	Person	Información del pagador
buyer	Person	Información del comprador
shipping	Person	Información del receptor
ipAddress	string[15]	Dirección IP desde la cual realiza la transacción el pagador
userAgent	string[255]	Agente de navegación utilizado por el pagador
additionalData	Attribute[]	Datos adicionales para ser almacenados con la transacción
credits	CreditConcept[]	Detalle de la dispersión a realizar

PSETransactionResponse

Estructura con la información de respuesta para la creación de una transacción.

Propiedades

Nombre	Tipo	Descripción
transactionID	int	Identificador único de la transacción en

		PlacetoPay
sessionID	string[32]	Identificador único de la sesión en PlacetoPay
returnCode	string[30]	Código de respuesta de la transacción, uno de los siguientes valores: SUCCESS FAIL_ENTITYNOTEXISTSORDISABLED FAIL_BANKNOTEXISTSORDISABLED FAIL_SERVICENOTEXISTS FAIL_INVALIDAMOUNT FAIL_INVALIDSOLICITDATE FAIL_BANKUNREACHEABLE FAIL_NOTCONFIRMEDBYBANK FAIL_CANNOTGETCURRENTCYCLE FAIL_ACCESSDENIED FAIL_TIMEOUT FAIL_DESCRIPTIONNOTFOUND FAIL_EXCEEDEDLIMIT FAIL_TRANSACTIONNOTALLOWED FAIL_RISK FAIL_NOHOST FAIL_NOTALLOWEDBYTIME FAIL_ERRORINCREDITS
trazabilityCode	string[40]	Código único de seguimiento para la operación dado por la red ACH
transactionCycle	int	Ciclo de compensación de la red
bankCurrency	string[3]	Moneda aceptada por el banco acorde a ISO 4217
bankFactor	float	Factor de conversión de la moneda
bankURL	string[255]	URL a la cual remitir la solicitud para iniciar la interfaz del banco, sólo disponible cuando returnCode = SUCCESS
responseCode	int	Estado de la operación en PlacetoPay [0 = FAILED, 1 = APPROVED, 2 = DECLINED, 3 = PENDING]
responseReasonCode	string[3]	Código interno de respuesta de la operación en PlacetoPay

responseReasonText	string[255]	Mensaje asociado con el código de respuesta de la operación en PlacetoPay

TransactionInformation

Estructura con la respuesta a una solicitud de información de transacción.

Propiedades

Nombre	Tipo	Descripción
transactionID	int	Identificador único de la transacción en PlacetoPay
sessionID	string[32]	Identificador único de la sesión en PlacetoPay
reference	string[32]	Referencia única de pago
requestDate	string	Fecha de solicitud o creación de la transacción acorde a ISO 8601
bankProcessDate	string	Fecha de procesamiento de la transacción acorde a ISO 8601
onTest	boolean	Indicador de si la transacción es en modo de pruebas o no
returnCode	string[30]	Código de respuesta de la transacción, uno de los siguientes: SUCCESS FAIL_INVALIDTRAZABILITYCODE FAIL_ACCESSDENIED FAIL_INVALIDSTATE FAIL_INVALIDBANKPROCESSINGDATE FAIL_INVALIDAUTHORIZEDAMOUNT FAIL_INCONSISTENTDATA FAIL_TIMEOUT FAIL_INVALIDVATVALUE FAIL_INVALIDTICKETID FAIL_INVALIDSOLICITEDATE FAIL_INVALIDAUTHORIZATIONID FAIL_TRANSACTIONNOTALLOWED FAIL_ERRORINCREDITS

		FAIL_EXCEEDEDLIMIT
trazabilityCode	string[40]	Código único de seguimiento para la operación dado por la red ACH
transactionCycle	int	Ciclo de compensación de la red
transactionState	string[20]	Información del estado de la transacción [OK, NOT_AUTHORIZED, PENDING, FAILED]
responseCode	int	Estado de la operación en PlacetoPay
responseReasonCode	string[3]	Código interno de respuesta de la operación en PlacetoPay
responseReasonText	string[255]	Mensaje asociado con el código de respuesta de la operación en PlacetoPay

Ejemplos

A continuación se muestran algunos ejemplos de generación de una fecha en formato ISO 8601 y de la generación del hash en SHA1.

Ejemplo de generación de fecha ISO 8601

A continuación se muestra como se da formato a la fecha actual para generar una cadena en ISO 8601.

PHP

```
$seed = date('c');
```

C# .NET

```
using System;
using System.Globalization;

String seed = DateTime.UtcNow.ToString("s",
        System.Globalization.CultureInfo.InvariantCulture);
```

JAVA

```
import java.util.Date;
import java.text.SimpleDateFormat;

Date now = new Date();
SimpleDateFormat isoDate = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssZ");
String seed = isoDate.format(now);
```

Ejemplo de generación del Hash

A continuación se presenta como se puede generar el Hash en diferentes lenguajes de programación. Este Hash es el que se remite como tranKey en la estructura de autenticación.

PHP

```
$hashString = sha1($seed
   . $tranKey, false);
```

C# .NET

```
using System.Text;
using System.Security.Cryptography;

private string GetSHA1(string text)
{
        ASCIIEncoding UE = new ASCIIEncoding();
        byte[] hashValue;
        byte[] message = UE.GetBytes(text);
        SHA1 hashString = new SHA1CryptoServiceProvider();
        string hex = "";
        hashValue = hashString.ComputeHash(message);
        foreach (byte x in hashValue) {
                hex += String.Format("{0:x2}", x);
        }
        return hex;
}

string hashString = GetSHA1(seed
        + tranKey);
```

JAVA

```
import java.security.MessageDigest;
import java.util.Formatter;

private static String byteToHex(final byte[] hash)
{
        Formatter formatter = new Formatter();
        for (byte b : hash)
        {
                formatter.format("%02x", b);
        }
        String result = formatter.toString();
```

```
        formatter.close();
        return result;
}

private static String GetSHA1(String password)
{
        String sha1 = "";
        try
        {
                MessageDigest crypt = MessageDigest.getInstance("SHA-1");
                crypt.reset();
                crypt.update(password.getBytes("UTF-8"));
                sha1 = byteToHex(crypt.digest());
        }
        catch(UnsupportedEncodingException e)
        {
                e.printStackTrace();
        }
        return sha1;
}

string hashString = GetSHA1(seed
        + tranKey);
```