MIGRACIÓN API 1.2

transbank.

Enero 2024

Introducción

Este documento detalla el proceso de migración necesario para adaptar tu integración a la API versión 1.2 de Transbank.

En él, encontrarás tablas con requisitos mínimos de lenguaje y versiones de SDKs, así como consideraciones esenciales para asegurar una transición exitosa.
Se abordan cambios operativos en el retorno al comercio y se proporcionan pautas sobre la estructura de llamadas a métodos para distintos lenguajes y SDKs.

Esta guía busca facilitar la actualización de tu integración, asegurando su compatibilidad con la última versión del API de Transbank. Es posible que necesites adaptaciones adicionales en tu software, el enfoque de este documento es únicamente para la integración con los productos de Transbank.

Índice

Proceso de migración

Integración directa con la API

Si realizaste la integración de forma directa con la API de Transbank, debes considerar los siguientes puntos para realizar una correcta migración:

- El retorno al comercio desde Transbank ahora se realiza **mediante el método GET**, anteriormente era por POST.
- Actualiza la versión de API en el endpoint del producto que estés usando a la v 1.2.

Producto	Endpoint
Webpay Plus	/rswebpaytransaction/api/webpay/v1.2/transactions
Webpay Plus Mall	/rswebpaytransaction/api/webpay/v1.2/transactions
Oneclick Mall	/rswebpaytransaction/api/oneclick/v1.2/inscriptions
Transacción completa	/rswebpaytransaction/api/webpay/v1.2/transactions
Transacción completa Mall	/rswebpaytransaction/api/webpay/v1.2/transactions

Integración usando un SDK oficial

Si realizaste tu integración utilizando un SDK oficial, debes tener varias consideraciones, en la versión 1.2 ahora el retorno al comercio desde Transbank es realizado mediante el **método GET**, revisa la versión del SDK a utilizar, la versión mínima del lenguaje soportado y la llamada a los métodos.

Un cambio significativo que se ha realizado en la forma de trabajar con los SDK es que se han reemplazado las clases estáticas por clases instanciables.

Antes de comenzar el proceso de migración de tu integración, asegúrate de que tu entorno cumple con los requisitos mínimos del lenguaje que estás utilizando.

La versiones de lenguajes mínimas requeridas son:

Lenguaje	Versión mínima requerida
Java	1.8
.NET	.NET Framework 4.5.2 .NET Standard 2.0
PHP	7.4
Node.js	8+
Python	3.4
Ruby	2.4+

Las versiones mínimas por SDK de Transbank que operan con la versión 1.2 del API son:

Lenguaje	Versión del SDK Transbank a instalar
Java	3.0.2
.NET	5.0.0
PHP	4.0.0
Node.js	3.0.2
Python	3.0.1
Ruby	2.0.1

Nota importante: Si tu SDK tiene una versión superior a la indicada, no es necesario actualizar tu integración para operar con la versión 1.2 del API.

Uso del los SDK por producto

Las llamadas a los métodos fueron actualizadas. En la tabla encontrarás el detalle de los cambios que debes implementar en tu integración por producto y lenguaje de programación.

Puedes apoyarte en los links de la documentación y proyectos de ejemplo disponibles en transbankdevelopers.cl

Webpay Plus

Java

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=java

Instalación del SDK	`Reemplaza el archivo POM.xml:`
	`<dependency><groupId>com.github.transbankdevelopers</groupId> <artifactId>transbank-sdk-java</artifactId> <version>3.0.2</version> </dependency>`

Crear una transacción	`import cl.transbank.webpay.webpayplus.responses.WebpayPlusTransactionCreateResponse;`
	`WebpayPlus.Transaction tx = new WebpayPlus.Transaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionCreateResponse response = tx.create(buyOrder, sessionId, amount, returnUrl);`

Confirmar una transacción	`WebpayPlus.Transaction tx = new WebpayPlus.Transaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionCommitResponse response = tx.commit(token);`

Consultar el estado de una transacción	`WebpayPlus.Transaction tx = new WebpayPlus.Transaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionStatusResponse response = tx.status(token);`

Reversar o anular una transacción	`WebpayPlus.Transaction tx = new WebpayPlus.Transaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionRefundResponse response = tx.refund(token, amount);`

Capturar una transacción	`WebpayPlus.Transaction tx = new WebpayPlus.Transaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionCaptureResponse response = tx.capture(token, buyOrder, authorizationCode, amount);`

.NET

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=csharp
Actualiza la versión del paquete	`PM> Update-Package TransbankSDK -version 5.0.0`

Crear una transacción	`using Transbank.Webpay.WebpayPlus;`
	
	`var tx = new Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = tx.Create(buyOrder, sessionId, amount, returnUrl);`

Confirmar una transacción	`using Transbank.Webpay.WebpayPlus;`
	
	`var tx = new Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = tx.Commit(token);`

Consultar el estado de una transacción	`using Transbank.Webpay.WebpayPlus;`
	
	`var tx = new Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = tx.Status(token);`

Reversar o anular una transacción	`using Transbank.Webpay.WebpayPlus;`
	
	`var tx = new Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = tx.Refund(token, refundAmount);`

Capturar una transacción	`using Transbank.Webpay.WebpayPlus;`
	
	`var tx = new Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_DEFERRED, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = tx.Capture(token, captureAmount);`

PHP

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=php
Actualiza la librería con composer	`composer update transbank/transbank-sdk:^4.0`

Crear una transacción	`use Transbank\Webpay\WebpayPlus\Transaction;`
	
	`$transaction = new Transaction();`
	`$response = $transaction->create($buy_order, $session_id, $amount, $return_url);`

Confirmar una transacción	`use Transbank\Webpay\WebpayPlus\Transaction;`
	
	`$response = (new Transaction)->commit($token);`

Consultar el estado de una transacción	`use Transbank\Webpay\WebpayPlus\Transaction;`
	
	`$response = (new Transaction)->status($token);`

Reversar o anular una transacción	`use Transbank\Webpay\WebpayPlus\Transaction;`
	
	`$response = (new Transaction)->refund($token, $amount);`

Capturar una transacción	`use Transbank\Webpay\WebpayPlus\Transaction;`
	
	`$response = (new Transaction)->capture($token, $buyOrder, $authCode, $amount);`

Node.js

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=javascript
Actualiza la librería con npm	`npm update transbank-sdk@3.0.2`

Crear una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.create(buyOrder, sessionId, amount, returnUrl);`

Confirmar una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.commit(token);`

Consultar el estado de una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.status(token);`

Reversar o anular una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.refund(token, amount);`

Capturar una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_DEFERRED, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.capture(token, buyOrder, authorizationCode, captureAmount);`

Python

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=python
Actualiza la librería con pip	`pip install --upgrade transbank-sdk==3.0.1`

Crear una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.create(buy_order, session_id, amount, return_url)`

Confirmar una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.commit(token)`

Consultar el estado de una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.status(token)`

Reversar o anular una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.refund(token, amount)`

Capturar una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.capture(token=token, buy_order=buy_order, authorization_code=authorization_code, capture_amount=amount)`

Ruby

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=ruby
Actualiza la librería con gem	`gem install transbank-sdk -v 2.0.1`

Crear una transacción	`@tx = Transbank::Webpay::WebpayPlus::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS)`
	`@resp = @tx.create( buy_order: buy_order, session_id: session_id, amount: amount, return_url: return_url )`

Confirmar una transacción	`@tx = Transbank::Webpay::WebpayPlus::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS)`
	`@resp = @tx.commit(token: @token)`

Consultar el estado de una transacción	`@tx = Transbank::Webpay::WebpayPlus::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS)`
	`@resp = @tx.status(token: @token)`

Reversar o anular una transacción	`@tx = Transbank::Webpay::WebpayPlus::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS)`
	`@resp = @tx.refund(token: @token, amount: @amount)`

Capturar una transacción	`@tx = Transbank::Webpay::WebpayPlus::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_DEFERRED)`
	`@resp = @tx.capture( token: @token, buy_order: @buy_order, authorization_code: @auth_code, amount: @amount )`

Webpay Plus Mall

Java

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=java#webpay-plus-mall

Instalación del SDK	`Reemplaza el archivo POM.xml:`
	
	`<dependency><groupId>com.github.transbankdevelopers</groupId> <artifactId>transbank-sdk-java</artifactId> <version>3.0.2</version> </dependency>`

Crear una transacción	`MallTransactionCreateDetails mallDetails = MallTransactionCreateDetails.build() .add(amountMallOne, commerceCodeMallOne, buyOrderMallOne) .add(amountMallTwo, commerceCodeMallTwo, buyOrderMallTwo);`
	
	`WebpayPlus.MallTransaction tx = new WebpayPlus.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final WebpayPlusMallTransactionCreateResponse response = tx.create(buyOrder, sessionId, returnUrl, mallDetails);`

Confirmar una transacción	`WebpayPlus.MallTransaction tx = new WebpayPlus.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final WebpayPlusMallTransactionCommitResponse response = tx.commit(token);`

Consultar el estado de una transacción	`WebpayPlus.MallTransaction tx = new WebpayPlus.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final WebpayPlusMallTransactionStatusResponse response = tx.status(token);`

Reversar o anular una transacción	`WebpayPlus.MallTransaction tx = new WebpayPlus.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final WebpayPlusTransactionRefundResponse response = tx.refund(token, childBuyOrder, childCommerceCode, amount);`

.NET

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=csharp#webpay-plus-mall
Actualiza la versión del paquete	`PM> Update-Package TransbankSDK -version 5.0.0`

Crear una transacción	`using Transbank.Webpay.WebpayPlus;`
	`var transactions = new List<TransactionDetail>();`
	`transactions.Add(new TransactionDetail( amountMallOne, commerceCodeMallOne, buyOrderMallOne ));`
	`transactions.Add(new TransactionDetail( amountMallTwo, comerceCodeMallTwo, buyOrderMallTwo ));`
	`var tx = new MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	`var result = tx.Create(buyOrder, sessionId, returnUrl, transactions);`

Confirmar una transacción	`using Transbank.Webpay.WebpayPlus;`
	`var tx = new MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`

	`var response = tx.Commit(token);`

Consultar el estado de una transacción	`using Transbank.Webpay.WebpayPlus;`
	`var tx = new MallTransaction(new`
	`Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL,`
	`IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	`var response = tx.Status(token);`

Reversar o anular una transacción	`using Transbank.Webpay.WebpayPlus;`
	`var tx = new MallTransaction(new`
	`Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL,`
	`IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	`var response = tx.Refund(token, buyOrder, commerceCode, amount);`

Capturar una transacción	`var tx = new MallTransaction(new`
	`Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL_DEFERRED,`
	`IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	`var response = tx.Capture(token, commerceCode, buyOrder,`
	`authorizationCode, captureAmount);`

PHP

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=php#webpay-plus-mall
Actualiza la librería con composer	`composer update transbank/transbank-sdk:^4.0`

Crear una transacción	`use Transbank\Webpay\WebpayPlus\MallTransaction;`
	`$transaction_details = [`
	`  [`
	`      "amount" => 10000,`
	`      "commerce_code" => 597055555536,`
	`      "buy_order" => "ordenCompraDetalle1234"`
	`  ],`
	`  [`
	`     "amount" =>12000,`
	`     "commerce_code" => 597055555537,`
	`     "buy_order" => "ordenCompraDetalle4321"`
	`  ],`
	`];`
	
	`$response = (new MallTransaction)->create($buy_order, $session_id, $return_url, $transaction_details);`

Confirmar una transacción	`use Transbank\Webpay\WebpayPlus\MallTransaction;`
	`$response = (new MallTransaction)->commit($token);`

Consultar el estado de una transacción	`use Transbank\Webpay\WebpayPlus\MallTransaction;`
	`$response = (new MallTransaction)->status($token);`

Reversar o anular una transacción	`$response = (new MallTransaction)->refund($token, $buy_order, $commerce_code, $amount);`

Capturar una transacción	`use Transbank\Webpay\WebpayPlus\MallTransaction;`
	`$response = (new MallTransaction)->capture($childCommerceCode, $token, $buyOrder, $authorizationCode, $captureAmount);`

Node.js

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=javascript#webpay-plus-mall
Actualiza la librería con npm	`npm update transbank-sdk@3.0.2`

Crear una transacción	`const TransactionDetail = require("transbank-sdk").TransactionDetail;`
	`let details = [ new TransactionDetail( amount, commerceCode, buyOrder ), new TransactionDetail( amount2, commerceCode2, buyOrder2 ), ];`
	`const tx = new WebpayPlus.MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.create( buyOrder, sessionId, returnUrl, details );`

Confirmar una transacción	`const tx = new WebpayPlus.MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.commit(token);`

Consultar el estado de una transacción	`const tx = new WebpayPlus.MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.status(token);`

Reversar o anular una transacción	`const tx = new WebpayPlus.MallTransaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.refund(token, buyOrder, commerceCode, amount);`

Capturar una transacción	`const tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS_DEFERRED, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.capture(token, buyOrder, authorizationCode, captureAmount);`

Python

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=python#webpay-plus-mall
Actualiza la librería con pip	`pip install --upgrade transbank-sdk==3.0.1`

Crear una transacción	`transaction_details = MallTransactionCreateDetails( amount_child_1, commerce_code_child_1, buy_order_child_1 ).add( amount_child_2, commerce_code_child_2, buy_order_child_2 )`
	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = tx.create( buy_order=buy_order, session_id=session_id, return_url=return_url, details = transaction_details, )`

Confirmar una transacción	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = tx.commit(token)`

Consultar el estado de una transacción	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = tx.status(token)`

Reversar o anular una transacción	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = tx.refund(token, child_buy_order, child_commerce_code, amount)`

Capturar una transacción	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.WEBPAY_PLUS_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	resp = tx.capture(token=token, capture_amount=amount, commerce_code=child_commerce_code, buy_order=buy_order, authorization_code=authorization_code)

Ruby

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/webpay-plus?l=ruby#webpay-plus-mall
Actualiza la librería con gem	gem install transbank-sdk -v 2.0.1
Crear una transacción	transaction_details = [{ amount: 10000, commerce_code: 597055555536, buy_order: "ordenCompraDetalle1234" }, { amount: 12000, commerce_code: 597055555537, buy_order: "ordenCompraDetalle4321" },] @tx = Transbank::Webpay::WebpayPlus::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_MALL) @resp = @tx.create(buy_order: buy_order, session_id: session_id, return_url: return_url, details: transaction_details)
Confirmar una transacción	@tx = Transbank::Webpay::WebpayPlus::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_MALL) @resp = @tx.commit(token: @token)

Consultar el estado de una transacción	`@tx = Transbank::Webpay::WebpayPlus::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_MALL)`
	`@resp = @tx.status(token: @token)`

Reversar o anular una transacción	`@tx = Transbank::Webpay::WebpayPlus::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_MALL)`
	`@resp = @tx.refund(token: @token, buy_order: @child_buy_order, child_commerce_code: @child_commerce_code, amount: @amount)`

Capturar una transacción	`@tx = Transbank::Webpay::WebpayPlus::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::WEBPAY_PLUS_MALL_DEFERRED)`
	`@resp = @tx.capture(child_commerce_code: @child_commerce_code, token: @token, buy_order: @buy_order, authorization_code: @authorization_code, capture_amount: @capture_amount)`

Oneclick Mall

Java

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=java

Instalación del SDK	`Reemplaza el archivo POM.xml:`
	
	`<dependency><groupId>com.github.transbankdevelopers</groupId>`
	`<artifactId>transbank-sdk-java</artifactId> <version>3.0.2</version>`
	`</dependency>`

Crear una inscripción	`String username = "nombre_de_usuario";`
	`String email = "nombre_de_usuario@gmail.com";`
	`String response_url = "https://callback/resultado/de/inscripcion";`
	
	`Oneclick.MallInscription inscription = new`
	`Oneclick.MallInscription(new`
	`WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL,`
	`IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final OneclickMallInscriptionStartResponse response =`
	`inscription.start(username, email, response_url);`

Confirmar una inscripción	`String tbk_token = "token_tbk";`
	
	`Oneclick.MallInscription inscription = new`
	`Oneclick.MallInscription(new`
	`WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL,`
	`IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final OneclickMallInscriptionFinishResponse response =`
	`inscription.finish(tbk_token);`

Eliminar una inscripción	String username = "nombre_de_usuario"; }
	String tbkUser = "tbkUserRetornadoPorInscriptionFinish";
	
	Oneclick.MallInscription inscription = new
	Oneclick.MallInscription(new
	WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL,
	IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	inscription.delete(tbkUser, username);

Autorizar una transacción	String username = "nombre_de_usuario";
	String tbkUser = "tbkUserRetornadoPorInscriptionFinish"; }
	String buyOrder = String.valueOf(new
	Random().nextInt(Integer.MAX_VALUE));
	double amountOne = 10000;
	String MallOneCommerceCode = "597055555542";
	String buyOrderMallOne = String.valueOf(new
	Random().nextInt(Integer.MAX_VALUE));
	int installmentNumberOne = 3;
	double amountTwo = 50000;
	String MallTwoCommerceCode = "597055555543";
	String buyOrderMallTwo = String.valueOf(new
	Random().nextInt(Integer.MAX_VALUE));
	int installmentNumberTwo = 3;
	
	MallTransactionCreateDetails details =
	MallTransactionCreateDetails.build() .add(amountOne,
	MallOneCommerceCode, buyOrderMallOne, installmentNumberOne)
	.add(amuntTwo, MallTwoCommerceCode, buyOrderMallTwo,
	installmentNumberTwo); Oneclick.MallTransaction
	
	tx = new Oneclick.MallTransaction(new
	WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL,
	IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final OneclickMallTransactionAuthorizeResponse response =
	tx.authorize(username, tbkUser, buyOrder, details);

Consultar el estado de una transacción	Oneclick.MallTransaction tx = new Oneclick.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final OneclickMallTransactionStatusResponse response = tx.status(buyOrder);

Reversar o anular una transacción	String buyOrder = "buyOrderIndicadoEnTransactionAuthorize";
	String childCommerceCode = "childCommerceCodeIndicadoEnTransactionAuthorize";
	String childBuyOrder = "childBuyOrderIndicadoEnTransactionAuthorize";
	double amount = 10000; Oneclick.MallTransaction
	
	tx = new Oneclick.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final OneclickMallTransactionRefundResponse response = tx.refund(buyOrder, childCommerceCode, childBuyOrder, amount);

Capturar una transacción	Oneclick.MallTransaction tx = new Oneclick.MallTransaction(new WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final OneclickMallTransactionCaptureResponse response = tx.capture(childCommerceCode, childBuyOrder, authorizationCode, amount);

.NET

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=csharp
Actualiza la versión del paquete	`PM> Update-Package TransbankSDK -version 5.0.0`

Crear una inscripción	`var username = "nombre_de_usuario";`
	`var email = "nombre_de_usuario@gmail.com";`
	`var response_url = "https://callback/resultado/de/inscripcion";`
	
	`var ins = new MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = ins.Start(userName, email, returnUrl);`

Confirmar una inscripción	`var token = "tbkToken";`
	
	`var ins = new MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = ins.Finish(tbk_token);`

Eliminar una inscripción	`var username = "nombre_de_usuario";`
	`var tbkUser = "tbkUserRetornadoPorInscriptionFinish";`
	`var ins = new MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));`
	
	`var response = ins.Delete(tbkUser, username);`

Autorizar una transacción	var username = "nombre_de_usuario";
	var tbkUser = "tbkUserRetornadoPorInscriptionFinish";
	var buyOrder = RandomString(10);
	var childCommerceCode = "597055555542";
	var childBuyOrder = RandomString(10);
	var amount = Decimal.Parse(Request.Form["amount"]);
	var installmentsNumber = 1; List<PaymentRequest>
	
	details = new List<PaymentRequest>(); details.Add(new PaymentRequest(childCommerceCode, childBuyOrder, amount, installmentsNumber));
	
	var tx = new MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));
	
	var result = tx.Authorize(userName, tbkUser, buyOrder, details);

Consultar el estado de una transacción	var tx = new MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));
	
	var result = tx.Status(buyOrder);

Reversar o anular una transacción	var buyOrder = "buyOrderIndicadoEnTransactionAuthorize";
	var childCommerceCode = "childCommerceCodeIndicadoEnTransactionAuthorize";
	var childBuyOrder = "childBuyOrderIndicadoEnTransactionAuthorize";
	var amount = 10000;
	
	var tx = new MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));
	
	var result = tx.Refund(buyOrder, childCommerceCode, childBuyOrder, amount);

Capturar una transacción	var tx = new MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, WebpayIntegrationType.Test));

	`var result = tx.Capture(ChildcommerceCode, ChildbuyOrder,`
	`authorizationCode, amount);`

PHP

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=php
Actualiza la librería con composer	`composer update transbank/transbank-sdk:^4.0`

Crear una inscripción	`use Transbank\Webpay\Oneclick\MallInscription;`
	
	`$username = "nombre_de_usuario";`
	`$email = "nombre_de_usuario@gmail.com";`
	`$response_url = "https://callback/resultado/de/inscripcion";`
	
	`$response = (new MallInscription)->start($username, $email,`
	`$response_url);`

Confirmar una inscripción	`use Transbank\Webpay\Oneclick\MallInscription;`
	
	`$tbk_token = $_GET['TBK_TOKEN'];`
	`$response = (new MallInscription)->finish($tbk_token);`
	`$tbkUser = $response->getTbkUser();`

Eliminar una inscripción	`use Transbank\Webpay\Oneclick\MallInscription;`
	
	`$username = 'nombre_de_usuario';`
	`$tbkUser = 'tbkUserRetornadoPorInscriptionFinish'; }`
	`$response = (new MallInscription)->delete($tbkUser, $username);`

Autorizar una transacción	`use Transbank\Webpay\Oneclick\MallTransaction;`
	`use Transbank\Webpay\Oneclick;`
	
	`$parentBuyOrder = rand(100000, 999999999);`
	`$details = [["commerce_code" => Oneclick::DEFAULT_CHILD_COMMERCE_CODE_1, "buy_order" => rand(100000, 999999999), "amount" => 50000, "installments_number" => 1], ["commerce_code" => Oneclick::DEFAULT_CHILD_COMMERCE_CODE_2, "buy_order" => rand(100000, 999999999), "amount" => 20000, "installments_number" => 1]];`
	
	`$response = (new MallTransaction)->authorize($username, $tbkUser, $parentBuyOrder, $details);`

Consultar el estado de una transacción	`use Transbank\Webpay\Oneclick\MallTransaction;`
	
	`$response = (new MallTransaction)->status($buyOrder);`

Reversar o anular una transacción	`use Transbank\Webpay\Oneclick\MallTransaction;`
	
	`$buyOrder = "buyOrderIndicadoEnTransactionAuthorize";`
	`$childCommerceCode = "childCommerceCodeIndicadoEnTransactionAuthorize";`
	`$childBuyOrder = "childBuyOrderIndicadoEnTransactionAuthorize";`
	`$amount = 10000;`
	
	`$response = (new MallTransaction)->refund($buyOrder, $childCommerceCode, $childBuyOrder, $amount);`

Capturar una transacción	`use Transbank\Webpay\Oneclick\MallTransaction;`
	
	`$response = (new MallTransaction)->capture($commerce_code, $buy_order, $authorization_code, $amount);`

Node.js

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=javascript
Actualiza la librería con npm	`npm update transbank-sdk@3.0.2`

Crear una inscripción	`const username = 'nombre_de_usuario';`
	`const email = 'nombre_de_usuario@email.com';`
	`const responseUrl = 'https://callback/resultado/de/inscripcion';`
	
	`const ins = new Oneclick.MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await ins.start(username, email, responseUrl);`

Confirmar una inscripción	`const token = 'tbkToken'`
	`const ins = new Oneclick.MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await ins.finish(token);`

Eliminar una inscripción	`username = 'nombre_de_usuario';`
	`tbkUser = 'tbkUserRetornadoPorInscriptionFinish';`
	
	`const ins = new Oneclick.MallInscription(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await ins.delete(tbkUser, username);`

Autorizar una transacción	`const details = [ new TransactionDetail(amount, commerceCode, childBuyOrder), new TransactionDetail(amount2, commerceCode2, childBuyOrder2) ];`
	`const tx = new Oneclick.MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.authorize(userName, tbkUser, buyOrder, details);`

Consultar el estado de una transacción	`const tx = new Oneclick.MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.status(token);`

Reversar o anular una transacción	`const buyOrder = "buyOrderIndicadoEnTransactionAuthorize";`
	`const childCommerceCode = "childCommerceCodeIndicadoEnTransactionAuthorize";`
	`const childBuyOrder = "childBuyOrderIndicadoEnTransactionAuthorize";`
	`const amount = 10000;`
	`const tx = new Oneclick.MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.refund(buyOrder, childCommerceCode, buyOrderChild, amount);`

Capturar una transacción	`const tx = new Oneclick.MallTransaction(new Options(IntegrationCommerceCodes.ONECLICK_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.capture(commerceCode, buyOrder, authorizationCode, amount);`

Python

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=python
Actualiza la librería con pip	`pip install --upgrade transbank-sdk==3.0.1`

Crear una inscripción	`username = "nombre_de_usuario"`
	`email = "nombre_de_usuario@gmail.com"`
	`response_url = "https://callback/resultado/de/inscripcion"`
	`ins =`
	`MallInscription(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL`
	`, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = ins.start(username=user_name, email=email,`
	`response_url=response_url)`

Confirmar una inscripción	`tbk_token = "tbkToken"`
	`ins =`
	`MallInscription(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL`
	`, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = ins.finish(token=tbk_token)`

Eliminar una inscripción	`username = "nombre_de_usuario"`
	`tbkUser = "tbkUserRetornadoPorInscriptionFinish"`
	`ins =`
	`MallInscription(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL`
	`, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`ins.delete(tbk_user=tbkUser, username=username)`

Autorizar una transacción	username = "nombre_de_usuario"
	tbkUser = "tbkUserRetornadoPorInscriptionFinish"
	buy_order = str(random.randrange(1000000, 99999999))
	commerce_code1 = "597055555542"
	buy_order_child1 = str(random.randrange(1000000, 99999999))
	installments_number1 = 3
	amount1 = 10000
	commerce_code2 = "597055555543"
	buy_order_child2 = str(random.randrange(1000000, 99999999))
	installments_number2 = 4
	amount2 = 50000
	details = MallTransactionAuthorizeDetails(commerce_code1,
	buy_order_child1, installments_number1, amount1) \\
	.add(commerce_code2, buy_order_child2, installments_number2,
	amount2)
	tx =
	MallTransaction(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL
	, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.authorize(username= username, tbk_user= tbkUser,
	parent_buy_order= buy_order, details= details)

Consultar el estado de una transacción	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL , IntegrationApiKeys.WEBPAY, IntegrationType.TEST)) resp = tx.status(buy_order)
Reversar o anular una transacción	buy_order = "buyOrderIndicadoEnTransactionAuthorize" child_commerce_code = "childCommerceCodeIndicadoEnTransactionAuthorize" child_buy_order = "childBuyOrderIndicadoEnTransactionAuthorize" amount = 10000 tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL , IntegrationApiKeys.WEBPAY, IntegrationType.TEST)) resp = tx.refund(buy_order, child_commerce_code, child_buy_order, amount)

Capturar una transacción	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.ONECLICK_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	`resp = tx.capture(child_commerce_code, child_buy_order, authorization_code, capture_amount)`

Ruby

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/oneclick?l=ruby
Actualiza la librería con gem	`gem install transbank-sdk -v 2.0.1`

Crear una inscripción	`@username = "nombre_de_usuario"`
	`@email = "nombre_de_usuario@gmail.com"`
	`@response_url = "https://callback/resultado/de/inscripcion"`
	`@ins = Transbank::Webpay::Oneclick::MallInscription.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL) }`
	
	`@resp = @ins.start(username: @username, email: @email, response_url: @response_url)`

Confirmar una inscripción	`@tbk_token = "tbkToken";`
	`@ins = Transbank::Webpay::Oneclick::MallInscription.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL)`
	
	`@resp = @ins.finish(token: @tbk_token)`

Eliminar una inscripción	`@username = "nombre_de_usuario"`
	`@tbkUser = "tbkUserRetornadoPorInscriptionFinish"`
	`@ins = Transbank::Webpay::Oneclick::MallInscription.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL)`
	
	`@resp = @ins.delete(tbk_user: @tbkUser, username: @username)`

Autorizar una transacción	@username = "nombre_de_usuario" @tbkUser = "tbkUserRetornadoPorInscriptionFinish" @buy_order = "12345" + Time.now.to_i.to_s @details = [{ commerce_code: "597055555542", buy_order: "abcdef" + Time.now.to_i.to_s, amount: 10000, installments_number: 3 }, { commerce_code: "597055555543", buy_order: "abcdef" + Time.now.to_i.to_s, amount: 50000, installments_number: 3 }] @tx = Transbank::Webpay::Oneclick::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL) @resp = @tx.authorize(username: @username, tbk_user: @tbkUser, parent_buy_order: @buy_order, details: @details)
Consultar el estado de una transacción	@tx = Transbank::Webpay::Oneclick::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL) @resp = @tx.status(buy_order: buy_order)
Reversar o anular una transacción	@buy_order = "buyOrderIndicadoEnTransactionAuthorize" @child_commerce_code = "childCommerceCodeIndicadoEnTransactionAuthorize" @child_buy_order = "childBuyOrderIndicadoEnTransactionAuthorize" @amount = 10_000 @tx = Transbank::Webpay::Oneclick::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL) @resp = @tx.refund(buy_order: @buy_order, child_commerce_code: @child_commerce_code, child_buy_order: @child_buy_order, amount: @amount)
Capturar una transacción	@tx = Transbank::Webpay::Oneclick::MallTransaction.new(::Transbank::Common::IntegrationCommerceCodes::ONECLICK_MALL_DEFERRED)

	@resp = @tx.capture(child_commerce_code: @commerce_code, child_buy_order: @buy_order, authorization_code: @authorization_code, amount: @capture_amount)

Transacción Completa

Java

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=java
Instalación del SDK	Reemplaza el archivo POM.xml: <dependency><groupId>com.github.transbankdevelopers</groupId> <artifactId>transbank-sdk-java</artifactId> <version>3.0.2</version> </dependency>

Crear una transacción	import cl.transbank.webpay.transaccioncompleta.responses.FullTransactionCreateResponse;
	
	String buyOrder = "Orden de compra de la transaccion";
	String sessionId = "Identificador del servicio unico de transacción";
	double amount = 10000;
	String cardNumber= "Numero de Tarjeta";
	String cardExpirationDate= "Fecha de expiracion en formato AA/MM";
	short cvv = 123;
	
	FullTransaction tx = new FullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	FullTransactionCreateResponse response = tx.create(buyOrder, sessionId, amount, cvv, cardNumber, cardExpirationDate);

Consulta de cuotas	import
	cl.transbank.webpay.transaccioncompleta.responses.FullTransactionCommitResponse;
	
	String token = "token obtenido como respuesta de la creacion de
	transaccion";
	int installmentsNumber = 10;
	
	FullTransaction tx = new FullTransaction(new
	WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA,
	IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final FullTransactionInstallmentResponse response =
	tx.installments(token, installmentsNumber);

Commit de transacción	import
	cl.transbank.webpay.transaccioncompleta.responses.FullTransactionCommitResponse;
	
	String token = "token obtenido como respuesta de la creacion de
	transaccion";
	int idQueryInstallments = 12345679;
	byte deferredPeriodIndex= 1;
	gracePeriod = false;
	
	FullTransaction tx = new FullTransaction(new
	WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA,
	IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final FullTransactionCommitResponse response = tx.commit(token,
	idQueryInstallments, deferredPeriodIndex, gracePeriod);

Consultar el estado de una transacción	import
	cl.transbank.webpay.transaccioncompleta.responses.FullTransactionStatusResponse;
	
	FullTransaction tx = new FullTransaction(new
	WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA,
	IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	final FullTransactionStatusResponse response = tx.status(token);

Reversar o anular una transacción	import cl.transbank.webpay.transaccioncompleta.responses.FullTransactionRefundResponse;
	FullTransaction tx = new FullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	final FullTransactionRefundResponse response = tx.refund(token, amount);

Capturar una transacción	FullTransaction tx = new FullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	final FullTransactionCaptureResponse response = tx.capture(token, buyOrder, authorizationCode, captureAmount);

.NET

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=csharp
Actualiza la versión del paquete	PM> Update-Package TransbankSDK -version 5.0.0

Crear una transacción	using Transbank.Webpay.TransaccionCompleta;
	
	var buy_order = "Orden de compra de la transaccion";
	var session_id = "Identificador del servicio unico de transacción";
	var amount = 10000;
	var card_number = "Numero de Tarjeta";
	var card_expiration_date = "Fecha de expiracion en formato AA/MM";
	var cvv = 123;
	
	var tx = new FullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	var response = tx.Create(buyOrder: buy_order, sessionId: session_id, amount: amount, cvv: cvv, cardNumber: card_number, cardExpirationDate: card_expiration_date);

Consulta de cuotas	using Transbank.Webpay.TransaccionCompleta;
	
	var token = "token obtenido como respuesta de la creacion de transaccion";
	var installments_number = 10;
	
	var tx = new FullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	
	var response = tx.Installments(token, installments_number);

Confirmar una transacción	using Transbank.Webpay.TransaccionCompleta;
	
	var token = "token obtenido como respuesta de la creacion de transaccion";
	var idQueryInstallments = 12345679;
	var deferredPeriodsIndex = 1;
	var gracePeriods = false;
	
	var tx = new FullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));

	`var response = tx.Commit(token, idQueryInstallments,`
	`deferredPeriodsIndex, gracePeriods);`

Consultar el estado de una transacción	`using Transbank.Webpay.TransaccionCompleta;`
	
	`var tx = new FullTransaction(new`
	`Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA,`
	`IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Status(token);`

Reversar o anular una transacción	`using Transbank.Webpay.TransaccionCompleta;`
	
	`var tx = new FullTransaction(new`
	`Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA,`
	`IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Refund(token, amount);`

Capturar una transacción	`var tx = new FullTransaction(new`
	`Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_DEFERRED,`
	`IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Capture(token, buyOrder, authorizationCode,`
	`captureAmount);`

PHP

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=php
Actualiza la librería con composer	`composer update transbank/transbank-sdk:^4.0`

Crear una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$transaction = new Transaction();`
	`$response = $transaction->create($buy_order, $session_id, $amount, $cvv, $card_number, $card_expiration_date);`

Consulta de cuotas	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$token = "token obtenido como respuesta de la creacion de transaccion";`
	`$installmentsNumber = 10;`
	`$transaction = new Transaction();`
	
	`$response = $transaction->installments($token, $installmentsNumber);`

Confirmar una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$token = "token obtenido como respuesta de la creacion de transaccion";`
	`$idQueryInstallments = 12345679;`
	`$deferredPeriodIndex= 1;`
	`$gracePeriod = false;`
	`$transaction = new Transaction();`
	
	`$response = $transaction->commit($token, $idQueryInstallments, $deferredPeriodIndex, $gracePeriod);`

Consultar el estado de una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$transaction = new Transaction();`
	`$transaction->status($token_ws);`

Reversar o anular una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$transaction = new Transaction();`
	`$transaction->refund($token, $amount);`

Capturar una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	
	`$transaction = new Transaction();`
	`$transaction->capture($token, $buyOrder, $authorizationCode, $captureAmount);`

Node.js

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=javascript
Actualiza la librería con npm	`npm update transbank-sdk@3.0.2`

Crear una transacción	`const TransaccionCompleta = require('transbank-sdk').TransaccionCompleta;`
	
	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.create(buyOrder, sessionId, amount, cvv, cardNumber, cardExpirationDate);`

Consulta de cuotas	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.installments(token, installmentsNumber);`

Confirmar una transacción	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.commit(token, idQueryInstallments, deferredPeriodIndex, gracePeriod);`

Consultar el estado de una transacción	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.status(token);`

Reversar o anular una transacción	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.refund(token, amount);`

Capturar una transacción	`const tx = new TransaccionCompleta.Transaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_DEFERRED, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.capture(token, buyOrder, authorizationCode, amount);`

Python

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=python
Actualiza la librería con pip	`pip install --upgrade transbank-sdk==3.0.1`

Crear una transacción	`from transbank.webpay.transaccion_completa.transaction import Transaction`
	
	`buy_order = 'Orden de compra de la transaccion '`
	`session_id = 'Identificador del servicio unico de transacción'`
	`amount = 10000;`
	`card_number = 'Numero de Tarjeta'`
	`cvv = 123`
	`card_expiration_date = 'Fecha de expiracion en formato AA/MM'`
	
	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`

	resp = tx.create(buy_order=buy_order, session_id=session_id,
	amount=amount, cvv=cvv, card_number=card_number,
	card_expiration_date=card_expiration_date)

Consulta de cuotas	from transbank.webpay.transaccion_completa.transaction
	import Transaction
	
	req = request.form
	token = request.form.get('token')
	installments_number = 10
	tx =
	Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLE
	TA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.installments(token=token,
	installments_number=installments_number)

Confirmar una transacción	from transbank.webpay.transaccion_completa.transaction
	import Transaction
	
	token = "token obtenido como respuesta de la creacion de transaccion"
	id_query_installments = 12345679
	deferred_period_index = 1
	grace_period = 'false'
	
	tx =
	Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLE
	TA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.commit(token=token,
	id_query_installments=id_query_installments,
	deferred_period_index=deferred_period_index,
	grace_period=grace_period)

Consultar el estado de una transacción	from transbank.webpay.transaccion_completa.transaction
	import Transaction
	
	tx =
	Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLE
	TA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.status(token=token)

Reversar o anular una transacción	`from transbank.webpay.transaccion_completa.transaction import Transaction`
	
	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.refund(token=token, amount=amount)`

Capturar una transacción	`tx = Transaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))`
	
	`resp = tx.capture(token, buy_order, authorization_code, capture_amount)`

Ruby

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=ruby
Actualiza la librería con gem	`gem install transbank-sdk -v 2.0.1`

Crear una transacción	`@buy_order = "Orden de compra de la transaccion"`
	`@session_id = "Identificador del servicio unico de transacción"`
	`@amount = 1000`
	`@card_number = "Numero de Tarjeta"`
	`@card_expiration_date = "Fecha de expiracion en formato AA/MM"`
	`@cvv = 123`
	
	`@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA)`

	@resp = @tx.capture(buy_order: @buy_order, session_id: @session_id, amount: @amount, cvv: @cvv, card_number: @card_number, card_expiration_date: @card_expiration_date)
Consulta de cuotas	token = "token obtenido como respuesta de la creacion de transaccion" installments_number = 10 @tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA) @resp = @tx.captureinstallments(token: token, installments_number: installments_number)

Confirmar una transacción	token = "token obtenido como respuesta de la creacion de transaccion"
	id_query_installments = 12345679
	deferred_period_index = 1
	grace_period = false
	
	@tx =
	Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA)
	
	@resp = @tx.commit(token: token, id_query_installments: id_query_installments, deferred_period_index: deferred_period_index, grace_period: grace_period)

Consultar el estado de una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA) @resp = @tx.status(token: token)
Reversar o anular una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA) @resp = @tx.refund(token: token, amount: amount)
Capturar una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_DEFERRED)

	`@resp = @tx.capture(token: token, buy_order: buy_order, authorization_code: authorization_code, capture_amount: capture_amount)`

Transacción Completa Mall

Java

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=java#transaccion-completa-mall

Instalación del SDK	`Reemplaza el archivo POM.xml:`
	`<dependency><groupId>com.github.transbankdevelopers</groupId> <artifactId>transbank-sdk-java</artifactId> <version>3.0.2</version> </dependency>`

Crear una transacción	`MallTransactionCreateDetails transactionDetails = MallTransactionCreateDetails.build() .add(amountMallOne, commerceCodeMallOne, buyOrderMallOne, installmentsNumberMallOne) .add(amountMallTwo, commerceCodeMallTwo, buyOrderMallTwo, installmentsNumberMallTwo);`
	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`MallFullTransactionCreateResponse response = tx.create(buyOrder, sessionId, cardNumber, cardExpirationDate, transactionDetails);`

Consulta de cuotas	`MallFullTransactionInstallmentsDetails installmentsDetails = MallFullTransactionInstallmentsDetails.build().add(commerceCode, buyOrder, installmentsNumber);`
	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final MallFullTransactionInstallmentsResponse response = tx.installments(token, installmentsDetails);`

Confirmar una transacción	`MallTransactionCommitDetails details = MallTransactionCommitDetails.build().add( commerceCode, buyOrder, idQueryInstallments, deferredPeriodIndex, gracePeriod);`
	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final MallFullTransactionCommitResponse response = tx.commit(token, details);`

Consultar el estado de una transacción	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final MallFullTransactionStatusResponse response = tx.status(tokenWs);`

Reversar o anular una transacción	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	`final MallFullTransactionRefundResponse response = tx.refund(token, buyOrder, childCommerceCode, amount);`

Capturar una transacción	`MallFullTransaction tx = new MallFullTransaction(new WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`final MallFullTransactionCaptureResponse response = tx.capture(token, commerceCode, buyOrder, authorizationCode, captureAmount);`

.NET

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=csharp#transaccion-completa-mall
Actualiza la versión del paquete	`PM> Update-Package TransbankSDK -version 5.0.0`

Crear una transacción	`using Transbank.Webpay.TransaccionCompletaMall;`
	
	`var transactionDetails = new List<CreateDetails>();`
	`transactionDetails.Add(new CreateDetails( amountMallOne, commerceCodeMallOne, buyOrderMallOne ));`
	
	`transactionDetails.Add(new CreateDetails( amountMallTwo, comerceCodeMallTwo, buyOrderMallTwo ));`
	
	`var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var details = new List<CreateDetails>();`

	details.Add(new CreateDetails(amountMallOne, commerceCodeMallOne, buyOrderMallOne)); details.Add(new CreateDetails(amountMallTwo, commerceCodeMallTwo, buyOrderMallTwo));
	var response = tx.Create(buyOrder: buy_order, sessionId: session_id, cardNumber: card_number, cardExpirationDate: card_expiration_date, details, cvv: cvv);

Consulta de cuotas	using Transbank.Webpay.TransaccionCompletaMall;
	var installmentsDetails = new List<MallInstallmentsDetails>(); installmentsDetails.Add(new CreateDetails(commerceCodeMallOne, buyOrderMallOne, installmentsNumberMallOne));
	installmentsDetails.Add(new CreateDetails(comerceCodeMallTwo, buyOrderMallTwo, installmentsNumberMallTwo));
	var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));
	var details = new List<CreateDetails>();
	details.Add(new CreateDetails(amountMallOne, commerceCodeMallOne, buyOrderMallOne));
	details.Add(new CreateDetails(amountMallTwo, commerceCodeMallTwo, buyOrderMallTwo));
	var response = tx.Installments(token, installmentsDetails);

Confirmar una transacción	using Transbank.Webpay.TransaccionCompletaMall;
	var transactionDetails = new List<MallCommitDetails>(); transactionDetails.Add(new MallCommitDetails(commerceCodeMallOne, buyOrderMallOne, idQueryInstallmentsOne, deferredPeriodIndexOne, gracePeriodOne));
	transactionDetails.Add(new MallCommitDetails(commerceCodeMallTwo, buyOrderMallTwo, idQueryInstallmentsTwo, deferredPeriodIndexTwo, gracePeriodTwo));

	`var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Commit( token, transactionDetails );`

Consultar el estado de una transacción	`using Transbank.Webpay.TransaccionCompletaMall;`
	
	`var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Status(token);`

Reversar o anular una transacción	`using Transbank.Webpay.TransaccionCompletaMall;`
	
	`var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Refund( token, buyOrder, childCommerceCode, amount );`

Capturar una transacción	`var tx = new MallFullTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST));`
	
	`var response = tx.Capture( token, childCommerceCode, buyOrder, authorizationCode, captureAmount );`

PHP

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=php
Actualiza la librería con composer	`composer update transbank/transbank-sdk:^4.0`

Crear una transacción	`use Transbank\TransaccionCompleta\MallTransaction;`
	`use Transbank\TransaccionCompleta\TransaccionCompleta;`
	
	`$transaction_details = [['amount' => 10000, 'commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_1, 'buy_order' => '123456789'], ['amount' => 12000, 'commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_2, 'buy_order' => '123456790'],]; $cardNumber = '4051885600446623'; $cardExpirationDate = '22/12'; $cvv = '123';`
	
	`$transaction = new MallTransaction();`
	
	`$response = $transaction->create( 'buyOrder1', // ordenCompra12345678 'sessionId' , // sesion1234564 $cardNumber, // 4239000000000000 $cardExpirationDate, // 22/10 $transaction_details, $cvv);`

Consulta de cuotas	`use Transbank\TransaccionCompleta;`
	`use Transbank\TransaccionCompleta\MallTransaction;`
	
	`$transaction = new MallTransaction(); $installments_details = [['commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_1, 'buy_order' => '123456789', 'installments_number' => 2], ['commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_2, 'buy_order' => '123456790', 'installments_number' => 2],];`
	
	`$response = $transaction->installments($token, $installments_details);`

Confirmar una transacción	`use Transbank\TransaccionCompleta;`
	`use Transbank\TransaccionCompleta\MallTransaction;`
	`$details = [['commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_1, 'buy_order' => '123456789', 'id_query_installments' => $firstInstallmentResponse->getIdQueryInstallments(), 'deferred_period_index' => null, 'grace_period' => false], ['commerce_code' => TransaccionCompleta::DEFAULT_MALL_CHILD_COMMERCE_CODE_2, 'buy_order' => '123456790', 'id_query_installments' => $secondInstallmentResponse->getIdQueryInstallments(), 'deferred_period_index' => null, 'grace_period' => false]];`
	`$transaction = new MallTransaction();`
	`$response = $transaction->commit($token, $details);`

Consultar el estado de una transacción	`use Transbank\TransaccionCompleta\MallTransaction;`
	`$response = (new MallTransaction)->status($token);`

Reversar o anular una transacción	`use Transbank\TransaccionCompleta\MallTransaction;`
	`(new MallTransaction)->refund( $token, $childBuyOrder, $childCommerceCode, $amount );`

Capturar una transacción	`use Transbank\TransaccionCompleta\Transaction;`
	`$transaction = new Transaction(); $transaction->capture($token, $buyOrder, $authorizationCode, $captureAmount);`

Node.js

Método	Código / Ejemplo
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=javascript#transaccion-completa-mall
Actualiza la librería con npm	`npm update transbank-sdk@3.0.2`

Crear una transacción	`const details = [ new TransactionDetail(amount, commerceCode, childBuyOrder), new TransactionDetail(amount2, commerceCode2, childBuyOrder2) ];`
	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.create( parentBuyOrder, sessionId, cardNumber, cardExpirationDate, details, cvv );`

Consulta de cuotas	`const details = [ new InstallmentDetail(childCommerceCode, childBuyOrder, installmentsNumber), new InstallmentDetail(childCommerceCode2, childBuyOrder2, installmentsNumber2) ];`
	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.installments( token, details );`

Confirmar una transacción	`let commitDetails = [ new CommitDetail(commerceCode, childBuyOrder), new CommitDetail(commerceCode2, childBuyOrder2) ];`
	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	`const response = await tx.commit( token, commitDetails );`

Consultar el estado de una transacción	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.status(token);`

Reversar o anular una transacción	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.refund( token, buyOrder, commerceCode, amount );`

Capturar una transacción	`const tx = new TransaccionCompleta.MallTransaction(new Options(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, Environment.Integration));`
	
	`const response = await tx.refund( token, commerceCode, buyOrder, authorizationCode, amount );`

Python

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=python#transaccion-completa-mall
Actualiza la librería con pip	`pip install --upgrade transbank-sdk==3.0.1`

Crear una transacción	`details = [ { 'commerce_code': 597055555552, 'buy_order': '123456789', 'amount': 10000 }, { 'commerce_code': 597055555553, 'buy_order': '123456790', 'amount': 12000 } ]`
	
	`tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST)) resp = tx.create( buy_order=buy_order, session_id=session_id,`

	card_number=card_number, card_expiration_date=card_expiration_date, details=details, cvv=None)

Consulta de cuotas	from transbank.webpay.transaccion_completa.mall_transaction import MallTransaction
	
	details = [{ 'commerce_code': 597055555552, 'buy_order': '123456789', 'installments_number': 2 }, { 'commerce_code': 597055555553, 'buy_order': '123456790', 'installments_number': 2 }]
	
	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.installments(token=token, details=details)

Confirmar una transacción	details = [{ "commerce_code": '597055555552', "buy_order": 'ordenCompra1234', "id_query_installments": 12, "deferred_period_index": 1, "grace_period": False }, { "commerce_code": '597055555553', "buy_order": 'ordenCompra12345', "id_query_installments": 12, "deferred_period_index": 1, "grace_period": False }]
	
	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.commit(token=token, details=details)

Consultar el estado de una transacción	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.status(token)

Reversar o anular una transacción	from transbank.webpay.transaccion_completa.mall_transaction import MallTransaction tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.refund(token=token, child_buy_order=child_buy_order, child_commerce_code=child_commerce_code, amount=amount)

Capturar una transacción	tx = MallTransaction(WebpayOptions(IntegrationCommerceCodes.TRANSACCION_COMPLETA_MALL_DEFERRED, IntegrationApiKeys.WEBPAY, IntegrationType.TEST))
	
	resp = tx.capture(token, child_commerce_code, child_buy_order, authorization_code, capture_amount)

Ruby

Método	**Código / Ejemplo**
Link TBK developers	https://www.transbankdevelopers.cl/documentacion/transaccion-completa?l=ruby#transaccion-completa-mall
Actualiza la librería con gem	gem install transbank-sdk -v 2.0.1
Crear una transacción	details = [{ amount: 10000, commerce_code: 597055555552, buy_order: '123456789' }, { amount: 12000, commerce_code: 597055555553, buy_order: '123456790' }] @buy_order = 'ordenCompra12345678' @session_id = 'sesion1234564' @card_number = 4239000000000000 @card_expiration_date = '22/10' @cvv = 123 @tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA) @resp = @tx.create(buy_order: @buy_order, session_id: @session_id, card_number: @card_number, card_expiration_date: @card_expiration_date, @details, cvv: @cvv)
Consulta de cuotas	installment_details = [{ commerce_code: 597055555552, buy_order: '123456789', installments_number: 2 }, { commerce_code: 597055555553, buy_order: '123456790', installments_number: 2 },]

	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_MALL) @resp = @tx.installments(token, installment_details)
Confirmar una transacción	details = [{ commerce_code: '597055555552', buy_order: 'ordenCompra1234', id_query_installments: 12, deferred_period_index: 1, grace_period: false }, { commerce_code: '597055555553', buy_order: 'ordenCompra12345', id_query_installments: 12, deferred_period_index: 1, grace_period: false }] @tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_MALL) @resp = @tx.commit(token: token, details: details)
Consultar el estado de una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_MALL) @resp = @tx.status(token)
Reversar o anular una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_MALL_DEFERRED) @resp = @tx.refund(token: token, buy_order: child_buy_order, commerce_code_child: child_commerce_code, amount: amount)
Capturar una transacción	@tx = Transbank::Webpay::TransaccionCompleta::Transaction.new(::Transbank::Common::IntegrationCommerceCodes::TRANSACCION_COMPLETA_MALL_DEFERRED) @resp = @tx.capture(token: token, commerce_code: commerce_code, buy_order: buy_order, authorization_code: authorization_code, amount: capture_amount)