

Pago con millas SDK

28 de febrero del 2020

Instalación

Para realizar la instalación se debe contar con composer previamente instalado. Luego de eso ejecutar el siguiente comando

```
composer install placetopay/mile-pay-sdk
```

Implementación

Para realizar la implementación de “Pago con millas SDK”, se deben importar lo siguiente.

```
use PlaceToPay\SDK\Client\PlaceToPayClientBuilder;  
use PlaceToPay\SDK\Exception\PlaceToPayException;
```

Luego se llama al constructor de la SDK, estableciendo todos los parámetros necesarios:

1. API URL
2. CLIENT ID
3. CLIENT SECRET
4. CALLBACK URL

```
try {  
  
    $placeToPayClient = PlaceToPayClientBuilder::builder()  
        ->withApiUrl("{YOUR API URL}")  
        ->withClientId("{YOUR CLIENT ID}")  
        ->withClientSecret("{YOUR CLIENT SECRET}")  
        ->withRedirectUrl("{YOUR CALLBACK URL}")  
        ->build();  
  
} catch (PlaceToPayException $sdkException) {  
    print $sdkException->getMessage();  
}
```

GetPoints

Método para obtener los puntos disponible del cliente y la tasa de conversión.

```
/*  
Get Points  
@params  
$merchantId: number ( número del cliente )  
  
@description  
-Endpoint para obtener la cantidad de millas disponibles  
*/  
$response = $placeToPayClient->getPoints($merchantId);  
/* RESPONSE:  
    {  
        "Data": {  
            "miles": 0,  
            "index_conversion": 0  
        },  
        "Message": "string"  
    }  
*/
```

LockPoints

Método para retener los puntos disponibles del cliente.

```
/*  
    Lock Points  
    @params  
    $milles: number  
  
    @description  
    -Endpoint para bloquear millas del cliente  
*/  
$response = $placeToPayClient->lockPoints($milles);  
/* RESPONSE:  
    {  
        "Data": {  
            "document_id": 0  
            "miles": 0,  
        },  
        "Message": "string"  
    }  
*/
```

DebitPoints

Método para debitar los puntos disponibles del cliente.

```
/*  
Debit Points  
@params  
$documentId: number  
  
@description  
-Endpoint para debitar la cantidad de millas disponibles  
*/  
$response = $placeToPayClient->debitPoints($documentId);  
/* RESPONSE:  
    {  
        "Data": {  
            "document_id": 0  
        },  
        "Message": "string"  
    }  
*/
```

LockAndDebitPoints

Método para retener y debitar los puntos disponibles del cliente.

```
/*  
    Lock Points  
    @params  
    $milles: number  
  
    @description  
    -Endpoint para bloquear millas del cliente  
*/  
  
$response = $placeToPayClient->lockAndDebitPoints($milles);  
/* RESPONSE:  
    {  
        "Data": {  
            "document_id": 0  
            "miles": 0,  
        },  
        "Message": "string"  
    }  
*/
```

CancelTransaction

Método para retener y debitar los puntos disponibles del cliente.

```
/*  
  
    Cancel Transaction  
  
    @params  
    $documentId: number  
  
    @description  
    -Endpoint para cancelar la transacción  
*/  
  
$response = $placeToPayClient->cancelTransaction($documentId);  
/* RESPONSE:  
    {  
        "Data": {  
            "document_id": 0  
        },  
        "Message": "string"  
    }  
*/
```

ReversePoint

Método para reversar la transacción

```
/*  
Reverse Point  
@params  
$documentId: number  
  
@description  
-Endpoint para devolver las millas debitadas.  
*/  
$response = $placeToPayClient->reversePoint($documentId);  
/* RESPONSE:  
    {  
        "Data": {  
            "document_id": 0  
        },  
        "Message": "string"  
    }  
*/
```