

Mercado Pago

 mercadopago.com.pe/developers/es/docs/checkout-api/integration-configuration/yape

Yape es una aplicación móvil que simplifica el proceso de transferencias bancarias. Los usuarios pueden realizar transacciones de manera fácil y rápida directamente desde su celular, después de asociar su tarjeta de débito MultiRed a la aplicación.

Para realizar una transacción con Yape, primero debes generar un token, necesario para la etapa de creación del pago. Esto puede ser hecho de dos formas: directamente a través de la API o utilizando el SDK JS de Mercado Pago.

En esta documentación encontrarás todas las etapas necesarias para realizar la configuración y las pruebas de integración con Yape de forma completa.

Con Checkout API, es posible ofrecer pagos a través de Yape utilizando el método de SDK JS para generar un token. Para ello, es necesario enviar los campos de celular y OTP (*One-time password* encontrado en la aplicación Yape). Con el token, es posible crear un pago.

Importante

Si deseas realizar la integración mediante API, avanza a la sección [Agregar formulario para captura de OTP y celular](#) y sigue las instrucciones.

Para ofrecer pagos con Yape, sigue los siguientes pasos.

La primera etapa del proceso de integración de pagos con Yape es la captura de los datos de OTP (*One-time password*) y celular para generar el token de pago. Esto se hace incluyendo la biblioteca MercadoPago.js en tu proyecto, seguida por el formulario para capturar los datos necesarios.

Utiliza el siguiente código para importar la biblioteca MercadoPago.js antes de añadir el formulario. Puedes hacerlo vía HTML o Bash.

- [html](#)
- [bash](#)

```
<body>
  <script src="https://sdk.mercadopago.com/js/v2"></script>
</body>
```

Las credenciales son claves únicas utilizadas para identificar una integración en tu cuenta. Se utilizan para capturar pagos en tiendas en línea y otras aplicaciones de forma segura.

Utiliza el código a continuación para configurar tu credencial, reemplazando el valor **YOUR_PUBLIC_KEY** por tu *Public key* de producción disponible en la aplicación creada en [Tus integraciones](#). Para más detalles, consulta [credenciales](#).

plain

```
const mp = new MercadoPago("YOUR_PUBLIC_KEY");
```

Para generar un token Yape, es necesario completar el campo OTP, que representa el código generado en la aplicación Yape, y el número de celular. Para capturar estos datos, utiliza el siguiente formulario HTML directamente en el proyecto:

html

```
<form id="form-checkout">
  <div>
    <label for="payerPhone">Phone Number</label>
    <input id="form-checkout__payerPhone" name="payerPhone" type="text" />
  </div>
  <div>
    <label for="payerOTP">OTP</label>
    <input id="form-checkout__payerOTP" name="payerOTP" type="text" />
  </div>
  <div>
    <button onclick="handleYape()">Create YAPE</button>
  </div>
</form>
```

Después de completar la inclusión del formulario y obtener los datos necesarios, genera el token de Yape utilizando MercadoPago.js o a través de la API.

Nota

El token es de un solo uso por compra y debe ser enviado durante la etapa de creación del pago.

Utiliza el método `mp.yape.create` para generar un token de Yape, según el siguiente código:

javascript

```
(async function handleYape () {
  const otp = document.getElementById("form-checkout__payerOTP").value;
  const phoneNumber = document.getElementById("form-checkout__payerPhone").value;
  const yapeOptions = {
    otp,
    phoneNumber
  };
  const yape = mp.yape(yapeOptions);
  const yapeToken = await yape.create();
  return yapeToken;
});
```

Otra manera de generar el token es a través de la API. Para ello, utiliza el siguiente cURL, completando los parámetros según lo detallado en la tabla a continuación.

Campo	Tipo	Descripción	Obligatorio/Opcional	Ejemplos/Posibles valores
phoneNumber	number	Número de celular del pagador obtenido en la etapa Capturar los datos de celular y OTP.	Obligatorio	123214234342
otp	number	Código único y numérico de 6 dígitos encontrado en la app de Yape. Este campo se obtiene en la etapa Capturar los datos de celular y OTP.	Obligatorio	123344
requestId	string	Campo generado automáticamente por el SDK JS. Debe enviarse sólo en las integraciones realizadas vía cURL.	Obligatorio para integraciones realizadas vía cURL	aaaaaaaa-bbbb-1ccc-8ddd-eeeeeeeeeeee

curl

```
curl --location 'https://api.mercadopago.com/platforms/pci/yape/v1/payment?public_key=<PUBLIC_KEY>' \
--header 'Content-Type: application/json' \
--data '{
  "phoneNumber": "123214234342",
  "otp": "123344",
  "requestId": "3127367123234234"
}
```

Ejemplo de la respuesta:

json

```
{
  "live_mode": true,
  "luhn_validation": null,
  "require_esc": null,
  "cloned": false,
  "cardholder": {
    "identification": {
      "number": null,
      "type": null,
      "subtype": null
    },
    "name": "yape"
  },
  "security_code_id": 8069792005119486812,
  "security_code_length": 6,
  "card_number_length": 9,
  "expiration_month": 5,
  "expiration_year": 2024,
  "card_present_id": null,
  "card_id": null,
  "client_id": 7775327892346559,
  "present": null,
  "id": "45d013f72bf42717a1625f4c508fc20f",
  "card_number_id": "FFTSHQTOSJTXGFVFGJKCBAIVOUISTFZBDRTQWLYJ",
  "creation_date": null,
  "due_date": null,
  "last_modified_date": null,
  "product_id": null,
  "trust_level": "unknown",
  "public_key": "APP_USR-352587ca-674b-4ae4-a348-8583ab39b4ac",
  "site_id": "MPE",
  "status": "active",
  "transaction_id": null,
  "trunc_card_number": null,
  "used_date": null,
  "bin": "111111",
  "version": 0,
  "client_header": null,
  "first_six_digits": "111111",
  "last_four_digits": "6789"
}
```

Después de añadir el formulario para capturar los datos del celular y del OTP y generar el token, debes crear el pago. Para hacerlo, envía el token proporcionado por el SDK de Mercado Pago JS y todos los datos necesarios al endpoint [/v1/payments](#). Estos datos incluyen 'transaction_amount', 'installments', 'payment_method_id' (específicamente **yape**), e la información del pagador. Si prefieres, también es posible hacer la solicitud utilizando uno de nuestros [SDKs backend](#).

El detalle de cada uno de los parámetros mencionados anteriormente, así como sus respectivos valores posibles, está descrito en la tabla a continuación.

Importante

Para esta etapa, al hacer la solicitud a través de la API o SDKs, es necesario enviar tu clave privada (*access_token* de producción).

Campo	Tipo	Descripción	Obligatorio/Opcional	Ejemplos/Posibles valores
-------	------	-------------	----------------------	---------------------------

Campo	Tipo	Descripción	Obligatorio/Opcional	Ejemplos/Posibles valores
<code>token</code>	string	Token proporcionado por el SDK JS de Mercado Pago. Para saber más, consulta la sección Generar token .	Obligatorio	"f8ae90c6a83e71d698d5ea927f851034"
<code>transaction_amount</code>	number	Valor de la transacción. Existe un límite máximo, que puede ser de S/500, S/900 o S/2000, configurado directamente en la propia aplicación Yape.	Obligatorio	2000
<code>description</code>	string	Título del producto.	Opcional	"Video game"
<code>installments</code>	number	Cantidad de cuotas. Al tratarse de un pago con tarjeta de débito, la cantidad de cuotas será 1.	Obligatorio	1
<code>payment_method_id</code>	string	<code>yape</code> para todos los casos.	Obligatorio	<code>yape</code>
<code>payer.email</code>	string	Email del pagador.	Obligatorio	" test_user_12345@gmail.com "

```

MercadoPagoConfig.setAccessToken("ENV_ACCESS_TOKEN");

PaymentClient client = new PaymentClient();

PaymentCreateRequest createRequest =
    PaymentCreateRequest.builder()
        .description("Titulo del producto")
        .installments(1)
        .payer(PaymentPayerRequest.builder()
            .email("test_user_123@testuser.com")
            .build())
        .paymentMethodId("yape")
        .token("ff8080814c11e237014c1ff593b57b4d")
        .transactionAmount(new BigDecimal("5000"))
        .build();

client.create(createRequest);

```

A continuación puedes ver un ejemplo de respuesta. Ten en cuenta que cierta información fue omitida para mostrar los campos más relevantes.

json

```

{
  "id": 74581527758,
  ...
  "payment_method_id": "yape",
  "payment_type_id": "debit_card",
  "payment_method": {
    "id": "yape",
    "type": "debit_card",
    "issuer_id": "12759",
    "data": {
      "routing_data": {
        "merchant_account_id": "462540702"
      }
    }
  },
  "status": "approved",
  "status_detail": "accredited",
  ...
}

```

Por ser una transacción con tarjeta de débito, los estados de pago posibles son **aprobado** o **rechazado**. Además, las políticas de reembolso y cancelación también se aplican.

Nota

En caso de que ocurra algún error al generar un pago, consulta la lista de posibles errores en la sección Referencia de API.

Es posible utilizar un OTP y números de celular de prueba para simular diferentes respuestas de pago en una transacción, sin la necesidad de usar números de celular y OTPs reales. Esto permite reproducir los estados mapeados en payments.

Para probar la integración, ingresa el OTP y uno de los números de celular listados en la tabla a continuación en el formulario de Checkout para simular escenarios de éxito y falla en la implementación.

Nota

Para probar la integración, debe usar las credenciales de prueba de su cuenta productiva. Para saber más, lee la [documentación](#).

Celular	OTP	Estado previsto en <code>payments</code>
111111111	123456	<code>approved</code>
111111112	123456	<code>cc_rejected_call_for_authorize</code>
111111113	123456	<code>cc_rejected_insufficient_amount</code>
111111114	123456	<code>cc_rejected_other_reason</code>
111111115	123456	<code>cc_rejected_card_type_not_allowed</code>
111111116	123456	<code>cc_rejected_max_attempts</code>
111111117	123456	<code>cc_rejected_bad_filled_security_code</code>
111111118	123456	<code>cc_rejected_form_error</code>

Los procedimientos para generar el token de Yape y crear el pago son los mismos: es necesario proporcionar el número de celular y el OTP de la tabla anterior. Si tienes dudas sobre cómo generar el token Yape o crear el pago, consulta las secciones [Generar token](#) y [Crear pago](#) respectivamente.