

Guia tecnica de integracion a Dpago via API

Documento para equipos tecnicos. Resume las URLs productivas, endpoints publicos de integracion, payload minimo, ejemplos de request y respuesta, y el flujo sugerido para salir a produccion.

URLs de produccion: API publica <https://api.dpago.com> y pasarela publica <https://pago.dpago.com>

1. Datos que el integrador necesita

Antes de implementar, el comercio debe solicitar a Dpago:

- `commerceld` del comercio.
- `platformId` de los medios de pago habilitados.
- `publicToken` del ambiente correspondiente.
- Confirmacion de si el flujo usara transaccion directa, link de pago o ambos.
- URL de callback o regla de confirmacion por consulta de estado.

2. Endpoint principal para crear pagos

Endpoint HTTP:

POST <https://api.dpago.com/transactions>

Headers minimos:

Content-Type: application/json

Token de validacion:

SHA256(reference + amount + publicToken)

Campo	Tipo	Descripcion
amount	number	Monto a cobrar.
description	string	Descripcion visible del pago.
reference	string	Referencia unica por transaccion.
customer	object	Incluye email, identification, name y phone.
token	string	Hash SHA256 del request.
platformId	number	Medio de pago habilitado en Dpago.
commerceld	number	Identificador del comercio en Dpago.

Ejemplo de payload JSON:

```
{
  "amount": 150000,
  "description": "Pago de pedido ORDER-1001",
  "reference": "ORDER-1001",
  "customer": {
    "email": "cliente@example.com",
    "identification": "1234567",
    "name": "Juan Perez",
    "phone": "0981123456"
  },
  "token": "HASH_SHA256_GENERADO",
```

```

    "platformId": 5,
    "commerceId": 10,
    "type": "common",
    "orderNumber": "ORDER-1001"
  }

```

Ejemplo cURL:

```

curl -X POST "https://api.dpago.com/transactions" \
-H "Content-Type: application/json" \
-d '{
  "amount": 150000,
  "description": "Pago de pedido ORDER-1001",
  "reference": "ORDER-1001",
  "customer": {
    "email": "cliente@example.com",
    "identification": "1234567",
    "name": "Juan Perez",
    "phone": "0981123456"
  },
  "token": "HASH_SHA256_GENERADO",
  "platformId": 5,
  "commerceId": 10
}'

```

3. Respuesta esperada al crear la transaccion

```

{
  "reference": "624561",
  "redirectUrl": "https://pago.dpago.com/...",
  "error": null,
  "qrInformation": null
}

```

Notas de integracion:

- `reference` de la respuesta es el ID interno de la transaccion en Dpago.
- `redirectUrl` es la URL que debe usarse para enviar al pagador a la pasarela.
- `qrInformation` puede venir informado en plataformas QR.
- Si la plataforma no usa redireccion, el comercio debe manejar la informacion alternativa devuelta.

4. Consulta de estado

```
GET https://api.dpago.com/transactions/:reference
```

Ejemplo:

```
GET https://api.dpago.com/transactions/624561
```

Estados visibles de transaccion:

- `pending`: pago pendiente de confirmacion.
- `completed`: pago confirmado.
- `refused`: pago rechazado.
- `reversed`: pago reversado.
- `incomplete`: pago incompleto.

5. Flujo sugerido de implementacion

1. Solicitar `commerceId`, `platformId` y `publicToken` de test.
2. Generar el hash `SHA256(reference + amount + publicToken)` desde el backend del comercio.
3. Ejecutar `POST /transactions` y guardar el `reference` interno devuelto.
4. Redirigir al usuario a `redirectUrl` o mostrar QR si aplica.
5. Confirmar el pago consultando `GET /transactions/:reference` o procesando callback.

6. 6. Repetir las pruebas en test antes de solicitar credenciales productivas.

Checklist tecnico previo a produccion
Generacion correcta del token SHA256.
Validacion de referencias unicas para evitar duplicados.
Manejo correcto de `redirectUrl` y estados `pending/completed/refused`.
Persistencia del `reference` interno devuelto por Dpago.
Validacion del flujo end-to-end en ambiente test.