**🔹 MercadoPagoConfig**

* **Qué hace:** Configura el token de acceso globalmente para usar la SDK.
* **Cómo se usa:**

MercadoPagoConfig.setAccessToken("ACCESS\_TOKEN");

* **Para qué:** Autenticar todas las operaciones que realices con la SDK.

**🔹 PreferenceClient**

* **Qué hace:** Crea y administra *preferencias de pago*.
* **Cómo se usa:**

PreferenceClient client = new PreferenceClient();

Preference preference = client.create(preferenceRequest);

* **Para qué:** Generar un link de pago personalizado.

**🔹 PreferenceRequest y PreferenceItemRequest**

* **Qué hace:** Representa el contenido de la preferencia (producto, precio, cantidad, etc.).
* **Cómo se usa:**

PreferenceItemRequest item = PreferenceItemRequest.builder()

.title("Producto")

.quantity(1)

.unitPrice(new BigDecimal(500))

.currencyId("ARS")

.build();

PreferenceRequest preferenceRequest = PreferenceRequest.builder()

.items(List.of(item))

.externalReference("12345") // ID de la transacción

.build();

* **Para qué:** Definir qué vas a cobrar y cómo.

**🔹 PaymentClient**

* **Qué hace:** Permite buscar, obtener, y hacer reembolsos de pagos.
* **Cómo se usa:**

PaymentClient client = new PaymentClient();

Payment payment = client.get(123456789L);

* **Para qué:** Consultar o interactuar con un pago realizado.

**🔹 Payment**

* **Qué hace:** Representa un pago ya realizado en Mercado Pago.
* **Cómo se usa:**

Payment payment = client.get(paymentId);

String estado = payment.getStatus();

String externalReference = payment.getExternalReference();

* **Para qué:** Verificar si un pago fue aprobado, rechazado, etc.

**🔹 RefundPaymentRequest y PaymentRefund**

* **Qué hace:** Ejecuta el reembolso total o parcial de un pago.
* **Cómo se usa:**

RefundPaymentRequest refundRequest = RefundPaymentRequest.builder().build();

PaymentRefund refund = client.refund(paymentId, refundRequest);

* **Para qué:** Devolverle la plata al comprador si el producto ya no está disponible.

**🟢 PreferenceBackUrls**

* **Qué hace:** Define a dónde redirigir al usuario luego del pago.
* **Cómo se usa:**

PreferenceBackUrls backUrls = PreferenceBackUrls.builder()

.success("https://tuapp.com/success")

.failure("https://tuapp.com/failure")

.pending("https://tuapp.com/pending")

.build();

* **Útil para:** Redireccionar al frontend cuando el usuario termina el pago.

**🟢 PreferencePaymentMethods**

* **Qué hace:** Te permite restringir medios de pago (por ejemplo, excluir efectivo).
* **Cómo se usa:**

PreferencePaymentMethods methods = PreferencePaymentMethods.builder()

.excludedPaymentTypes(List.of(

ExcludedPaymentType.builder().id("ticket").build()

))

.installments(1)

.build();

* **Útil para:** Permitir solo tarjeta, limitar cuotas, etc.

**🟢 MerchantOrderClient y MerchantOrder**

* **Qué hace:** Te permite gestionar la *orden del vendedor* (útil para varios pagos o seguimiento).
* **Cómo se usa:**

MerchantOrderClient orderClient = new MerchantOrderClient();

MerchantOrder order = orderClient.get(merchantOrderId);

* **Útil para:** Confirmar que el dinero fue liberado, ver pagos múltiples.

**🟢 WebhookNotification**

* **Qué hace:** Representa el contenido que Mercado Pago te envía a tu endpoint (webhook).
* **Cómo se usa:** Ya lo manejás como DTO (WebhookDTO), pero también podés mapear directamente usando las clases internas de la SDK.

**🟢 PaymentSearchRequest + client.search()**

* **Qué hace:** Buscar múltiples pagos con filtros.
* **Cómo se usa:**

PaymentSearchRequest searchRequest = PaymentSearchRequest.builder()

.externalReference("12345")

.build();

client.search(searchRequest);

* **Útil para:** Buscar todos los pagos de una transacción específica o de un usuario.

**Recomendaciones extras**

* ⚠️ **Agregá un dateOfExpiration o duración máxima del link** de pago si querés que caduque luego de un tiempo.
* 🛑 **Verificá en el webhook que el estado sea approved antes de marcar como pagado.**
* 🔄 **Implementá reembolso parcial** si ofrecés pago con más de un ítem.