PaymentGatewayAsyncAdapter Interface

Implement the interface to allow customers to process payments asynchronously.

Namespace

CommercePayments

Usage

Implementing an asynchronous adapter also requires the `processNotification` method fror GatewayNotificationResponse class.

Example

```
global with sharing class SampleAsyncAdapter
        implements commercepayments.PaymentGatewayAsyncAdapter,
        commercepayments.PaymentGatewayAdapter {

        global SampleAsyncAdapter() {
        }

        global commercepayments.GatewayResponse processRequest(
        commercepayments.paymentGatewayContext gatewayContext) {
        }

        global commercepayments.GatewayNotificationResponse processNotification(
        commercepayments.PaymentGatewayNotificationContext gatewayNotificationCont
        }
        }
```

- PaymentGatewayAsyncAdapter Methods
- PaymentGatewayAsyncAdapter Example Implementation

PaymentGatewayAsyncAdapter Methods

The following are methods for `PaymentGatewayAsyncAdapter`.

- processNotification(paymentGatewayNotificationContext)
  Entry point for processing notifications from payment gateways.

processNotification(paymentGatewayNotificationContext)

Entry point for processing notifications from payment gateways.

Signature

```
global commercepayments.GatewayNotificationResponse
processNotification(commercepayments.PaymentGatewayNotificationContext var1)
```

The `PaymentGatewayNotificationContext` object wraps all the information related to a gatev notification.

Return Value

Type: GatewayNotificationResponse

When the payment gateway sends a notification to the payments platform, the platform re with a `GatewayNotificationResponse` indicating whether the platform succeeded or failed at receiving the notification.

PaymentGatewayAsyncAdapter Example Implementation

This is a sample implementation of the `commercepayments.PaymentGatewayAsyncAdapter` interf

```
global with sharing class AdyenAdapter implements commercepayments.PaymentGatewayA
    global AdyenAdapter() {}

    global commercepayments.GatewayResponse processRequest(commercepayments.paymer
    }

    global commercepayments.GatewayNotificationResponse processNotification(commer
    }
}

commercepayments.RequestType requestType = gatewayContext.getPaymentRequestType()
if (requestType == commercepayments.RequestType.Capture) {
    req.setEndpoint('/pal/servlet/Payment/v52/capture');
    body = buildCaptureRequest((commercepayments.CaptureRequest)gatewayContext.get
} else if (requestType == commercepayments.RequestType.ReferencedRefund) {
    req.setEndpoint('/pal/servlet/Payment/v52/refund');
    body = buildRefundRequest((commercepayments.ReferencedRefundRequest)gatewayCon
}

req.setBody(body);
req.setMethod('POST');
commercepayments.PaymentsHttp http = new commercepayments.PaymentsHttp();
HttpResponse res = null;
try {
    res = http.send(req);
} catch(CalloutException ce) {
    commercepayments.GatewayErrorResponse error = new commercepayments.GatewayErro
    return error;
}

if ( requestType == commercepayments.RequestType.Capture) {
    response =  createCaptureResponse(res);
} else if ( requestType == commercepayments.RequestType.ReferencedRefund) {
    response =  createRefundResponse(res);
}
return response;

commercepayments.PaymentGatewayNotificationRequest notificationRequest = gatewayNo
Blob request = notificationRequest.getRequestBody();
Map<String, Object> jsonReq = (Map<String, Object>)JSON.deserializeUntyped(request
List<Object> notificationItems = (List<Object>)jsonReq.get('notificationItems');
Map<String, Object> notificationRequestItem =
    (Map<String, Object>)((Map<String, Object>)notificationItems[0]).get('Notifica
Boolean success = Boolean.valueOf(notificationRequestItem.get('success'));
String pspReference = (String)notificationRequestItem.get('pspReference');
String eventCode = (String)notificationRequestItem.get('eventCode');
Double amount = (Double)((Map<String, Object>)notificationRequestItem.get('amount

commercepayments.NotificationStatus notificationStatus = null;
if (success) {
```

```
        notification = new commercepayments.CaptureNotification();
    } else if ('REFUND'.equals(eventCode)) {
        notification = new commercepayments.ReferencedRefundNotification();
    }
    notification.setStatus(notificationStatus);
    notification.setGatewayReferenceNumber(pspReference);
    notification.setAmount(amount);

    commercepayments.NotificationSaveResult saveResult = commercepayments.Notificatior

    commercepayments.GatewayNotificationResponse gnr = new commercepayments.GatewayNot
    if (saveResult.isSuccess()) {
        system.debug('Notification accepted by platform');
    } else {
        system.debug('Errors in the result '+ Blob.valueOf(saveResult.getErrorMessage
    }
    gnr.setStatusCode(200);
    gnr.setResponseBody(Blob.valueOf('[accepted]'));
    return gnr;
```