Windows SDK - Migration Guide

Opening the Checkout using a webview

You can use a webview to open the checkout in your app in order to continue to allow your customers to make purchases in-app.

With webviews we recommend you to generate the checkout URL using the Pay link API. The reason for this is because Paypal opens in the same window with Pay Link API checkouts, creating less problems since Paypal opens on a separate pop up for paddle.js checkouts.

A basic example of a webview control in C# :

```
<UserControl x:Class="PaddleSDK.Checkout.CheckoutControl"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/ 2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:PaddleSDK.Checkout"
xmlns:web="clr-namespace:PaddleSDK.Web"
mc:Ignorable="d"
 DataContext="{Binding RelativeSource={RelativeSource Mode=Self}}"  MaxWidth="600"
MaxHeight="800"
 d:DesignHeight="450" d:DesignWidth="800">
 <Grid x:Name="mainGrid">
 <WindowsFormsHost>
 <web:WinFormsWebBrowser x:Name="browser"
Navigating="Browser_Navigating" DocumentCompleted="Browser_DocumentCompleted"/>
</WindowsFormsHost>
 </Grid>
</UserControl>
```

Note: Note the guide and lambda below rely on webhooks for license activation but you can also set a `return_url` when calling the pay link API with the `{checkout_hash}` and call the orders API to get the license details.

Licensing using webhooks

Please note: that we can export existing license details upon request.

One-off SDK products

To help you set this up quicker we've created a simple licensing Lambda on how to use our webhooks to generate licenses and validate them.

You can find the Lambda and accompanying PDF guide in a ZIP file which you can download from here.

If you want to remove access after a refund or chargeback we recommend you to listen to the `payment_refunded` and `payment_dispute_created` webhooks.

Subscriptions (without license generation)

For subscriptions the webhooks you need to listen to are different from the one-off lambda product example above.

See the guide on what webhooks you should listen to when using subscriptions.

New Subscription

1- Listen to the `subscription_payment_succeeded` webhooks where `initial_payment`= `1` and store the relevant user information (eg, `order_id`, `subscription_id`, `quantity`, `status`) and set the number of activations for the license based on the `subscription_plan_id` if different from plan to plan.

2 - Set the license expiry date based on the `next_bill_date` + the number of dunning retry days set in the Recover Settings page.

3 - Generate the license (code example provided in the licensing lambda) and deliver that to your customers. Alternatively you can always create an email address based activation.

Subscription renewal

Successful payments

Listen to the `subscription_payment_succeeded` webhooks where `initial_payment`= `0`, get the `subscription_id` and `next_bill_date` and update the license expiry date for that license.

Failed payments

When a subscription payment renewal fails we will try to charge according to the retry rules you've set in the Recover Settings page on Paddle's dashboard.

When all attempts have failed we will set the subscription status to Past Due, Paused or Cancelled according to what is set in the Recover Settings page.

Past Due flow

Listen to the `subscription_payment_failed` webhook to determine if the last payment attempt failed by looking at the `instalments` parameter.

Please note after the last attempt if the buyer decides to update the card details and the payment is successful we will fire the `subscription_payment_succeeded` webhook.

Subscription cancelation Flow

Listen to the `subscription_cancelled` webhooks get the `subscription_id` and `cancellation_effective_date` to set the expiry date to be the same as the `cancellation_effective_date`.

If the last payment is also refunded and you wish to revoke access immediately you can listen to the `subscription_payment_refunded` webhook to get the `subscription_id` and set the expiry date to the same day.

Subscription Paused Flow

Listen to the `subscription_updated` webhook where `status` = `paused` and set the expiry date to be the same as the `paused_from` date returned in the webhook.

Again if the last payment is also refunded and you wish to revoke access immediately you can listen to the `subscription_payment_refunded` webhook to get the `subscription_id` and set the expiry date to the same day.