

Technical Guide for Checkout.com Salesforce Order Management

Contents

Pre-Requirements	3
Installing the Application	3
Configuration of the Application	4
I. Create a Payment Gateway Provider	4
II. Create Named Credentials.....	5
III. Create Payment Gateway.....	6
IV. Checkout Payment Webhook Notification.....	7
V. Give Read Access to the Site Guest User	9
VI. Create Named Credentials for Notifications	11
VII. Webhook Fallback Settings.....	12
VIII. Checkout.com Account Configuration.....	13
IX. Klarna Settings	14

1 Pre-Requirements

To install Checkout.com package for Salesforce Order Management, you must be logged in as an administrator in Salesforce.com. The application can be installed in Professional, Enterprise, Force.com, Developer and Unlimited editions.

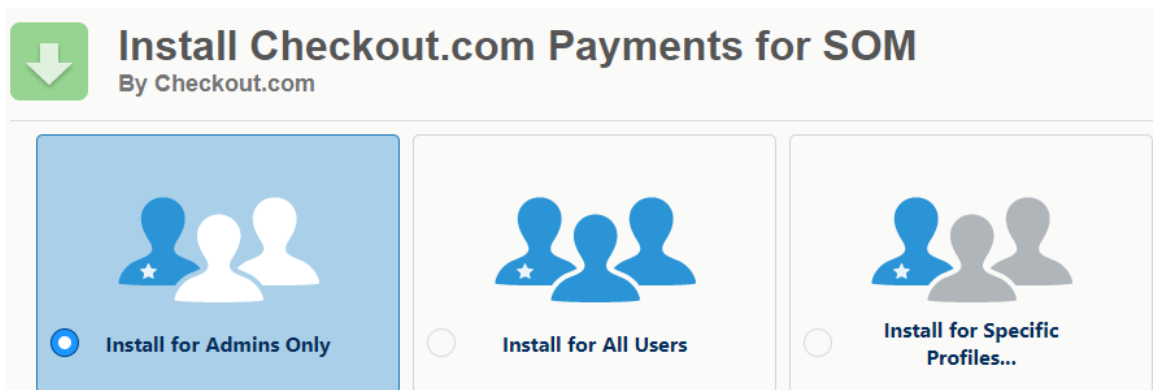
Important: Before installing the Checkout.com for Salesforce Order Management the following must be already configured:

- I. Order Management must be installed and configured, see [Salesforce Order Management implementation guide](#).

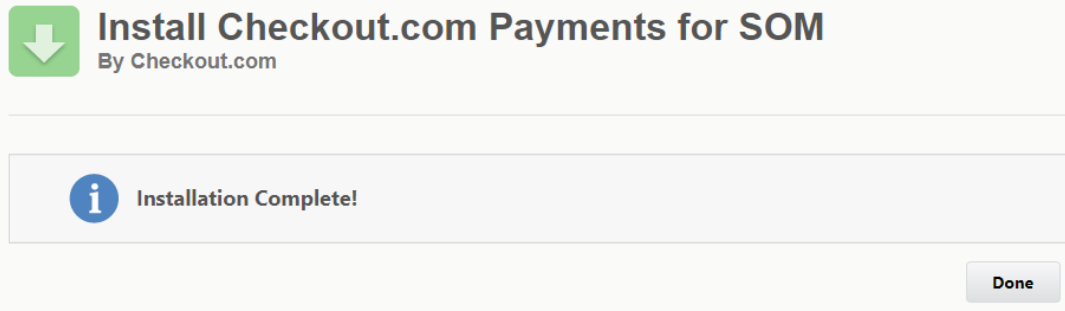
2 Installing the Application

Please follow the next steps to install the Checkout.com package:

1. Search for "Checkout.com for Salesforce Order Management" on AppExchange.
2. Click the "Get It Now" button on the AppExchange page.
3. Choose where you wish to install the product: Production or Sandbox.
4. To begin the installation, select "I have read and agree to the terms and conditions." and click "Confirm and Install!"
5. Enter your credentials.
6. In the next step you will be asked to choose a security level. After selecting your preference, click "Install" to continue the installation.



7. After the installation process has finished, you will see a message saying: "Install Complete". Click on the "Done" button.



3 Configuration of the Application

I. Create a Payment Gateway Provider

Creating a Payment Gateway Provider is currently NOT supported from the UI and should be done through [workbench](#) by connecting to your organization:

Using Workbench login with your org's credentials and follow the next steps:

1. Utilities -> REST Explorer
2. HTTP Method: POST
3. URL: /services/data/v51.0/composite
4. Request Body:

```
{
  "allOrNone": true,
  "compositeRequest": [{
    "method": "GET",
    "url":
"/services/data/v51.0/query/?q=SELECT+Id+FROM+ApexClass+WHERE+Name+=+'CheckoutAdapter'",
    "referenceId": "refCheckoutAdapter"
  },
  {
    "method": "POST",
    "url": "/services/data/v51.0/subjects/PaymentGatewayProvider",
    "referenceId": "refPaymentGatewayProvider",
    "body": {
      "ApexAdapterId": "@{refCheckoutAdapter.records[0].Id}",
      "DeveloperName": "Checkout_Payment_Provider ",
      "MasterLabel": "Checkout_Payment_Provider",
      "IdempotencySupported": "Yes",
      "Comments": "Comments"
    }
  }
]
}
```

5. Click Execute
6. Look for the success indicator - (status codes in the 200-299 range and ids which are populated).

workbench
info
queries
data
migration
utilities

REST Explorer

ANA LEVINTA AT DEVELOPMENT ORG ON API 51.0

Try the Salesforce APIs for Postman.

Choose an HTTP method to perform on the REST API service URI below:

☐ GET
☒ POST
☐ PUT
☐ PATCH
☐ DELETE
☐ HEAD
Headers
Reset
Up

Request Body

```

{
  "allOrNone": true,
  "compositeRequest": [
    {
      "method": "GET",
      "url": "/services/data/v51.0/query/?q=SELECT+Id+FROM+ApexClass+WHERE+Name+=+CheckoutAdapter'",
      "referenceId": "refCheckoutAdapter"
    },
    {
      "method": "POST",
      "url": "/services/data/v51.0/subjects/PaymentGatewayProvider",
      "referenceId": "refPaymentGatewayProvider",
      "body": {
        "ApexAdapterId": "@{refCheckoutAdapter.records[0].Id}",
        "DeveloperName": "Checkout_Payment_Provider ",
        "MasterLabel": "Checkout_Payment_Provider",
        "IdempotencySupported": "Yes",
        "Comments": "Comments"
      }
    }
  ]
}

```

Expand All | Collapse All | Show Raw Response

```

compositeResponse
├── [Item 1]
│   ├── body
│   │   ├── totalSize: 1
│   │   ├── done: true
│   │   └── records
│   ├── httpHeaders
│   │   ├── httpStatusCode: 200
│   │   └── referenceId: refCheckoutAdapter
│   └── [Item 2]
│       ├── body
│       │   ├── id: 0cJ09000000TncBEAW
│       │   ├── success: true
│       │   └── errors
│       ├── httpHeaders
│       │   ├── httpStatusCode: 201
│       │   └── referenceId: refPaymentGatewayProvider

```

II. Create Named Credentials

You will need to set up named credentials which is used in Checkout adapter.

1. Setup -> Security -> Named Credentials -> New Named Credentials
2. Fill the following information:

Field Name	Description/Value
Label	Checkout Payment
Name	Checkout_Payment  Important: The name must be <i>Checkout_Payment</i> for the application to work as expected.
URL(Production)	https://api.checkout.com
URL(Test)	https://api.sandbox.checkout.com

Identity Type	Anonymous
Authentication Protocol	No Authentication
Generate Authorization Header	Select the checkbox
Allow Merge Fields in HTTP Header	Select the checkbox
Allow Merge Fields in HTTP Body	Select the checkbox

3. Click Save

III. Create Payment Gateway

Follow the next steps to create a payment gateway in OMS:

1. App Launcher -> Payment Gateways -> New
2. Type "Checkout" in "Payment Gateway Name" field. (Or whatever you prefer)
3. Start typing "Checkout" in "Payment Gateway Provider". Find and select the Checkout payment gateway provider that we created previously.
4. Start typing "Checkout" in "Merchant Credential". Find and select Checkout named credentials.
5. Select "Active" for the "Status" field
6. For external Reference match with the payment processor id from SFCC
7. Click "Save".

New Payment Gateway

Payment gateway

* Payment Gateway Name

* Payment Gateway Provider

* Merchant Credential

* Status

Active

External Reference

Comments

Cancel

Save & New

Save

IV. Checkout Payment Webhook Notification

For allowing Checkout to send webhook notifications to OMS, a new site must be created.

Please follow the next steps to create a site:

1. Setup -> Sites
2. Register your **SiteDomain**

 **Note:** Pay attention to the chosen site name. Choose a site domain name which will be used for Checkout webhook notifications.

Sites

What is a Site?

Salesforce sites enables you to create public websites and applications that are directly integrated with your Salesforce.com organization—without requiring users to log in with a username and password. You can publicly your company's brand. Use sites to create public community sites to gather customer feedback, branded login and registration pages for your portals, Web forms for capturing leads, and so on.

Because sites are hosted on Salesforce servers, there are no data integration issues. And because sites are built on native Visualforce pages, data validation on collected information is performed automatically. You can al register your own branded domain or subdomain to redirect to your site.

Create multiple sites that appeal to different audiences and satisfy your company's various business needs. For example, a software company could create one site for new developers, another for customers, and a third f

Sample sites URLs

Sample Domain Name: MyCompany.eu40.force.com

MyCompany.eu40.force.com/customers

MyCompany.eu40.force.com/developers

MyCompany.eu40.force.com/partners

To get started, first register your company's Salesforce site domain. Your Salesforce site domain must be unique and must consist of only alphanumeric characters. Salesforce.com recommends using your company's nam

 You cannot modify your Salesforce site domain name after the registration process.

http:// -developer-edition.eu40.force.com

Success: The Salesforce site domain name "chk-notifications-developer-edition" is available.

☐ I have read and accepted the Salesforce [Sites Terms of Use](#)

3. Click on New for creating a new site
4. Set the following values:
 - Site Label: Checkout_Notifications
 - Site Name: automatically populated (Checkout_Notifications)
 - Site Description: Used for exposing REST services to Checkout Webhooks Notifications.
 - Active: checked
 - Active Site Home Page: set a home page (eg. AnswersHome)
 - Active: Guest Access to the Payments API



Site Edit Checkout_Notifications

Site Edit

Save Cancel

Site Label

Checkout_Notifications

Site Name

Checkout_Notifications

Site Description

Used for exposing REST services to Checkout Webhooks Notifications.

Site Contact

Ana Levinta

Default Record Owner

Ana Levinta

Default Web Address

http://chk-notification-developer-edition.eu40.force.com/

Active

☒

Active Site Home Page

AnswersHome [Preview]

Inactive Site Home Page

Site Template

SiteTemplate

Site Robots.txt

Site Favorite Icon

Analytics Tracking Code

URL Rewriter Class

Enable Feeds

☐

Clickjack Protection Level

Allow framing by the same origin only (Recommended)

Lightning Features for Guest Users

☒

Enable Content Sniffing Protection

☒

Enable Browser Cross Site Scripting Protection

☒

Referrer URL Protection

☒

Allow only required cookies for this site

☐ [More Details](#)

Redirect to custom domain

☒

Cache public Visualforce pages

☒ [\[More Info\]](#)

Guest Access to the Payments API

☒

Guest Access to the Support API

☐

Save Cancel

5. Click on site label
6. Open Public Access Settings

Sites (chk-notification-developer-edition.eu40.force.com) New

Action	Site Label	Site URL	Site Description	Active	Site Type	Last Modified By
Edit Deactivate Checkout_Notifications 1		https://chk-notification-developer-edition.eu40...	Used for exposing REST services to Checkout Webhooks Notifications.	✓	Force.com	Ana Levinta, 6/19/2021, 12:47 AM

Site Details
Checkout_Notifications Help for this Page

[Back to List: Sites](#)

Site Detail Edit Public Access Settings Login Settings URL Redirects Deactivate

Site Label	Site Description	Site Name	Site Contact
Checkout_Notifications	Used for exposing REST services to Checkout Webhooks Notifications.	Checkout_Notifications	Ana Levinta
Active	☒	Login	Not Allowed
Active Site Home Page	Anavertions (Preview)	Site Favorite Icon	
Inactive Site Home Page		Site Robots.txt	
Site Template	SiteTemplate (Preview)	Enable Feeds	☐
Analytics Tracking Code		URL Rewriter Class	
Clickjack Protection Level	Allow framing by the same origin only (Recommended)	Lightning Features for Guest Users	☒ 1
Enable Content Sniffing Protection	☒ 1	Enable Browser Cross Site Scripting Protection	☒ 1
Referrer URL Protection	☒ 1	Allow only required cookies for this site	☐ 1 More Details
Guest Access to the Payments API	☒ 1	Guest Access to the Support API	☐ 1
Default Record Owner	Ana Levinta 1	Redirect to custom domain	☒ 1
Cache public Visualforce pages	☒ 1 More Info		
Created By	Ana Levinta, 6/19/2021 11:39 PM	Last Modified By	Ana Levinta, 6/19/2021 12:47 AM

Edit Public Access Settings Login Settings URL Redirects Deactivate

7. Navigate to Apex Class Access

8. Add the Webhook Listener class: Give access to checkoutcom.CheckoutNotificationListener class.

SETUP
Profiles

Profile
Checkout_Notifications Profile

Edit Properties

[Profile Overview](#) > **Apex Class Access** ▼

Apex Class Access Save Cancel

Available Apex Classes	Enabled Apex Classes
- SiteLoginController - SiteLoginControllerTest - SiteRegisterController - SiteRegisterControllerTest - checkoutcom.BaseRequest - checkoutcom.CalloutUtility - checkoutcom.CheckoutAdapter - checkoutcom.CheckoutAdapterTest - checkoutcom.CheckoutNotificationHandler - checkoutcom.CheckoutNotificationHandlerTest - checkoutcom.CheckoutNotificationListenerTest - checkoutcom.CheckoutPaymentConfigQuery - checkoutcom.CheckoutPaymentConfigQueryTest - checkoutcom.CheckoutPaymentConfigTriggerHandler - checkoutcom.CheckoutPaymentConfigTriggerHandlerTest	checkoutcom.CheckoutNotificationListener

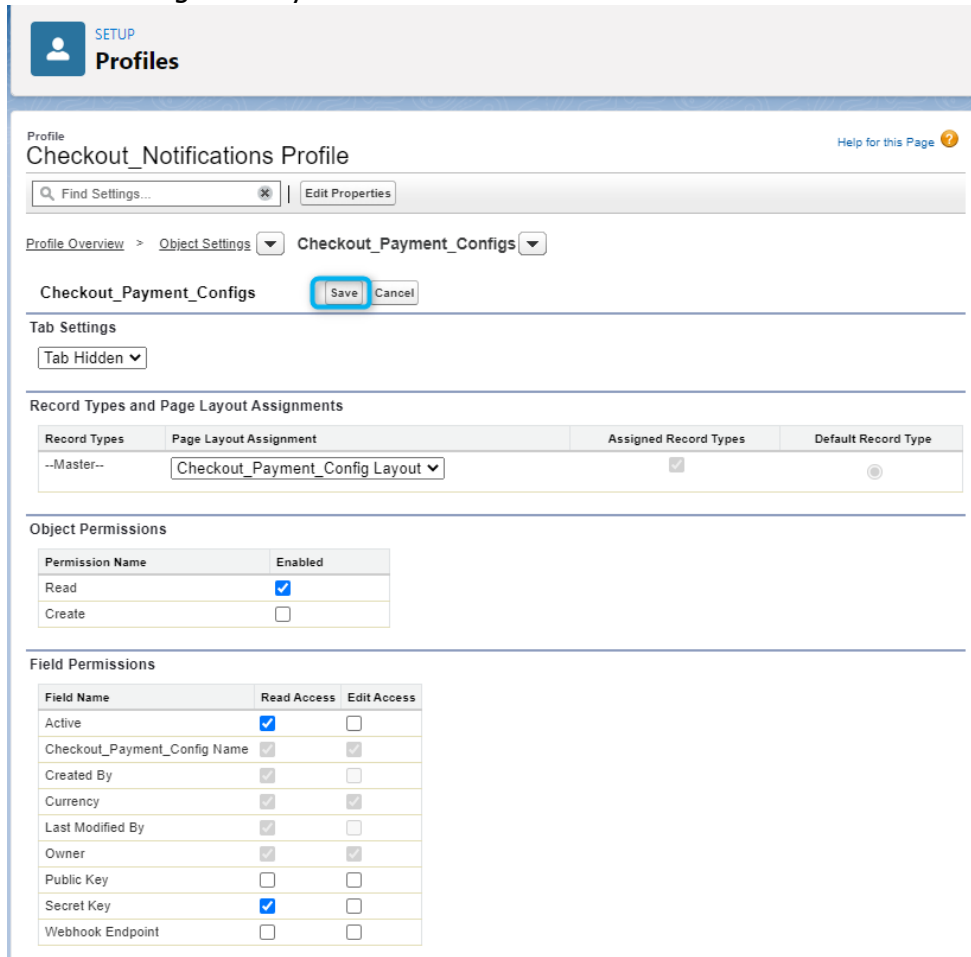
Add Remove

9. Click Save

V. Give Read Access to the Site Guest User

To give read access to the site guest user (webhook notifications) for the Checkout Payment follow the next steps:

1. On the site page click on Public Access Settings
2. Navigate to Object Settings
3. Click on *Checkout_Payment_Configs* Object Name
4. Click Edit
5. On Object Permissions, select 'Read'
6. On Field Permissions, give 'Read Access' for the 'Active' and 'Secret Key' fields
7. Leave the other settings as they are and click Save



Profile
Checkout_Notifications Profile

Find Settings... | Edit Properties

Profile Overview > Object Settings > Checkout_Payment_Configs

Checkout_Payment_Configs **Save** Cancel

Tab Settings
Tab Hidden

Record Types and Page Layout Assignments

Record Types	Page Layout Assignment	Assigned Record Types	Default Record Type
--Master--	Checkout_Payment_Config Layout	☒	☐

Object Permissions

Permission Name	Enabled
Read	☒
Create	☐

Field Permissions

Field Name	Read Access	Edit Access
Active	☒	☐
Checkout_Payment_Config Name	☐	☒
Created By	☐	☐
Currency	☐	☒
Last Modified By	☐	☐
Owner	☐	☒
Public Key	☐	☐
Secret Key	☒	☐
Webhook Endpoint	☐	☐

Since no authentication is used for the site, we must create a sharing rule for extending the access for the webhook notifications. Follow the next steps to do so:

1. Go to Setup -> Security -> Sharing Settings
2. Scroll down to 'Checkout_Payment_Config Sharing Rules'
3. Click New
4. Complete the fields with the following values:
 - i. Label -> Share Config with Webhook Notification
 - ii. Rule Name -> Share_Config_with_Webhook_Notification
 - iii. Description -> Sharing rule for webhook notifications
 - iv. Rule Type -> Guest user access, based on criteria
 - v. On Criteria area, for Field select Active, Operator equals, Value True
 - vi. Share with -> Checkout_Notifications Site Guest User
 - vii. Access Level -> Read Only
 - viii. Click Save

SETUP
Sharing Settings

Setup
Checkout_Payment_Config Sharing Rule

[Help for this Page](#) 

Use sharing rules to make automatic exceptions to your organization-wide sharing settings for defined sets of users.

Note: "Roles and subordinates" includes all users in a role, and the roles below that role.

You can use sharing rules only to grant wider access to data, not to restrict access.

Step 1: Rule Name
! = Required Information

Label
 Rule Name 
 Description

Step 2: Select your rule type

Rule Type
 ☐ Based on record owner
 ☐ Based on criteria
 ☒ Guest user access, based on criteria

Step 3: Select which records to be shared



This sharing rule grants access to guest users without login credentials. By modifying the default settings in accordance with these criteria, you're allowing immediate and unlimited access to all records matching these criteria to anyone accessing the site, even without logging in. To secure your site and its data from guest users, consider all the use cases and implications, and implement security controls that you think are appropriate for the sensitivity of your data. Salesforce isn't responsible for any exposure of your data to guest users related to this change from default settings.

Criteria	Field	Operator	Value	
	Active	equals	True	 AND
	--None--	--None--		AND
	--None--	--None--		AND
	--None--	--None--		AND
	--None--	--None--		AND

Add Filter Logic...

Additional Options
 ☐ Include records owned by high-volume users 

Step 4: Select the users to share with

Share with

Step 5: Select the level of access for the users

Access Level

VI. Create Named Credentials for Notifications

You will need to set up named credentials which is used for webhook listener.

1. Setup -> Named Credentials -> New Named Credentials
2. Fill the following information
 - ii. Label -> Checkout Notification
 - iii. Name -> Checkout_Notification

 **Important:** The name must be *Checkout_Notification* for the application to work as expected.

iv. URL ->

<https://{YourDomain}/services/data/v51.0/commerce/payments/notify?provider={CheckoutPaymentProviderId}>

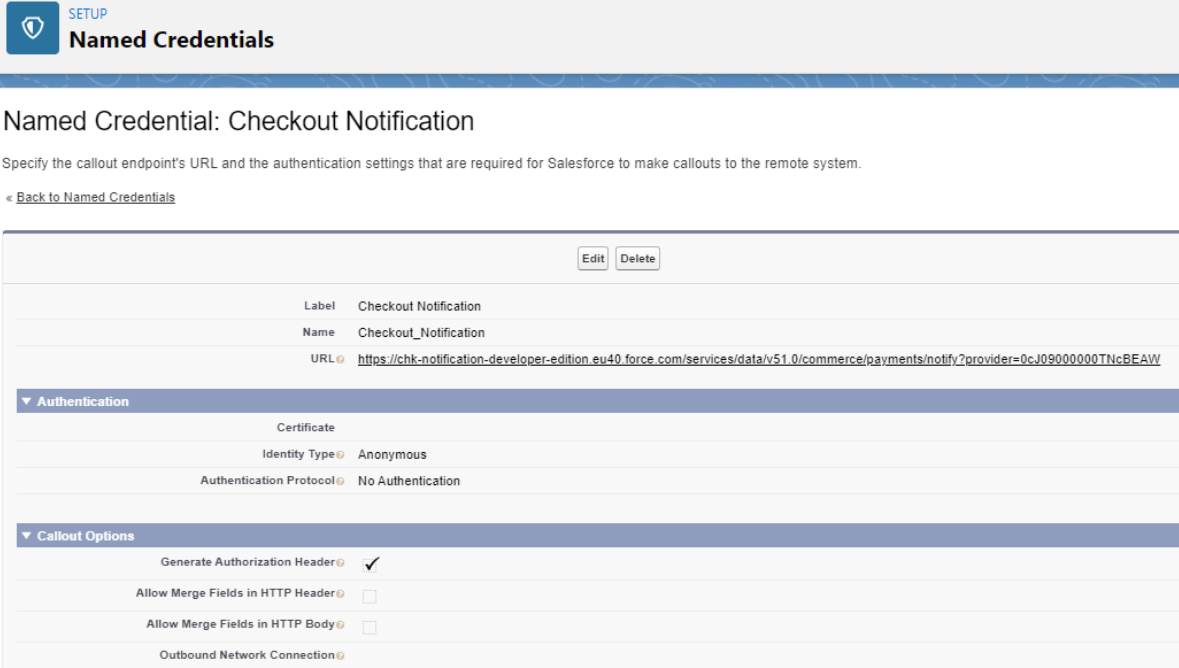
- *{YourDomain}* should be Site domain in your Org, the one you just created
- *{CheckoutPaymentProviderId}* should be the Id of payment gateway provider. You can find the Id from Workbench -> Queries -> SOQL Query -> *PaymentGatewayProvider*

v. Identity Type -> Anonymous

vi. Authentication Protocol -> No Authentication

vii. Select "Generate Authorization Header"

3. Click Save



VII. Webhook Fallback Settings

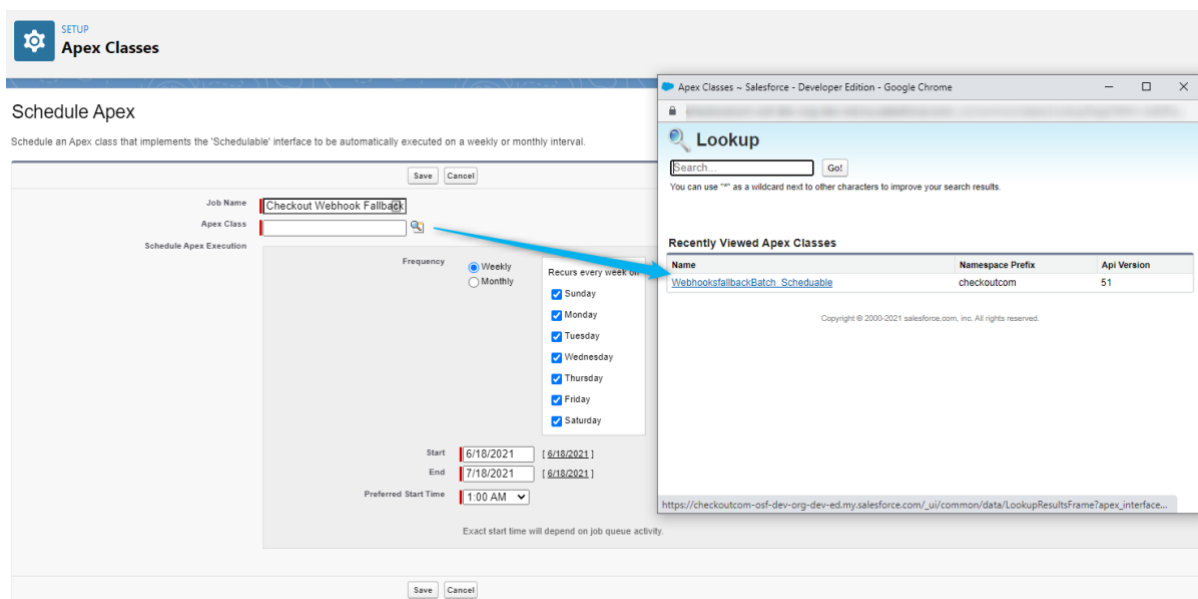
To ensure that the payments are processed even in case of a webhook fallback, a job can be scheduled to check the payment status in the Checkout Hub and update the payment in OMS. The check is done for the payments which are in Pending status in OMS and will be marked as processed if the payment is captured in the Checkout.com Hub.

Follow the next steps to schedule the job to run daily:

1. Go to Setup| Custom Code| Apex Classes
2. Click on Schedule Apex button
3. Specify the name of a class that you want to schedule, in our case:

WebhooksfallbackBatch_Scheduable

4. Specify how often the Apex class will run.
 - Weekly: specify one or more days of the week the job is to run (such as Monday and Wednesday).
 - Monthly: specify either the date the job is to run or the day (such as the second Saturday of every month.)
5. Specify the start and end dates for the Apex scheduled class. If you specify a single day, the job only runs once.
6. Specify a preferred start time. The exact time the job starts depends on service availability.
7. Click Save



VIII. Checkout.com Account Configuration

1. Go to App Launcher, search for Checkout_Payment_Config
2. Click New
3. Fill the following information:
 - i. Checkout_Payment_Config Name -> Checkout Settings (it can be anything)
 - ii. Webhook Endpoint -> use the endpoint for the site created [above](https://chk-notification-developer-edition.eu40.force.com/services/apexrest/checkoutcom/Checkout_Notifications):
https://chk-notification-developer-edition.eu40.force.com/services/apexrest/checkoutcom/Checkout_Notifications
 - iii. Secret Key: your Checkout.com secret key
 - iv. Public Key: your Checkout.com public key
 - v. Active: Checked

*Checkout_Payment_Config Name

Checkout Settings

Webhook Endpoint

https://chk-notification-developer-edition.eu40.force.c

Secret Key

*****_key

Public Key

*****_key

Active

☒

Cancel Save & New Save

 **Note:** When a Checkout Payment Config is created/updated with Active field set to true, in the background we subscribe to the Checkout.com webhook notifications using as URL the Webhook Endpoint. The events to which we are subscribing are those from Setup -> Custom Code -> Custom Metadata Types -> Webhook Event Types. By default, all the webhook events are marked as Subscribed True and should remain as they are for ensuring that the right webhook notifications will be processed.

IX. Klarna Settings

For supporting the Klarna settings into OMS, some configurations need to be checked and updated if is the case. These setting are used when capturing a payment having Klarna as payment method. To update them please follow the next steps:

1. Go to Setup -> Custom Code -> Custom Metadata Types
2. Click on Manage Records for *Klarna Setting* Label
3. Click Edit for the *Default* value
4. Check the settings and select/deselect based on your preferences

 **Important:** The Environment should be set as *Sandbox* if you are testing the package; as *Production* for going live.

 **SETUP**
Custom Metadata Types

Klarna Setting (Managed)

 This Klarna Setting is managed, meaning that you may only edit certain attributes. [Display More Information](#)

Klarna Setting Edit

SaveSave & NewCancel

Information

Label

Default

Klarna Setting Name

Default



Include Description

☒

Description Details 

Your order with Checkout.com

Include Products 

☒

Include Shipping Info 

☒

Include Shipping Delay

☐

Shipping Delay Value 

0

Environment 

Sandbox



SaveSave & NewCancel