

# Steps to Configure Radom Crypto Payments Salesforce B2C Plugin

## Step 1

Extract the zip file.

## Step 2

Add the int\_radomPaymentPlugin cartridge in your project and add update the cartridge path for your site from Administration > Sites > Manage Sites.

## Step 3

Add the path for radom.js in the checkout.isml

Path: templates/default/checkout/checkout.isml

```
<isscript>
  var assets = require('*/cartridge/scripts/assets.js');
  assets.addJs('/js/checkout.js');
  assets.addJs('/js/radom.js');
  assets.addCss('*/css/checkout/checkout.css');
</isscript>
```

## Step 4

add highlighted code in paymentOptionsTabs.isml

Path: templates/default/checkout/billing/paymentOptions/paymentOptionsTabs.isml

```
cartridges > int_radomPaymentPlugin > cartridge > templates > default > checkout > billing > paymentOptions > paymentOptionsTabs.isml
1 <isloop items="${pdict.order.billing.payment.applicablePaymentMethods}" var="paymentOption">
2   <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
3     <isinclude template="checkout/billing/paymentOptions/creditCardTab" />
4   </isif>
5   <isif condition="${paymentOption.ID === 'Radom'}">
6     <isinclude template="radom/checkout/radomTab" />
7   </isif>
8 </isloop>
```

## Step 5

add highlighted code in paymentOptionsContent.isml

Path: templates/default/checkout/billing/paymentOptions/paymentOptionsContent.isml

```
cartridges > app_storefront_base > cartridge > templates > default > checkout > billing > paymentOptions > paymentOptionsContent.isml
1 <isloop items="${pdict.order.billing.payment.applicablePaymentMethods}" var="paymentOption">
2   <isif condition="${paymentOption.ID === 'CREDIT_CARD'}">
3     <isinclude template="checkout/billing/paymentOptions/creditCardContent" />
4   </isif>
5   <isif condition="${paymentOption.ID === 'Radom'}">
6     <isinclude template="radom/checkout/radomcontent" />
7   </isif>
8 </isloop>
```

## Step 6

add highlighted code in paymentOptionsSummary.isml

Path: templates/default/checkout/billing/paymentOptions/paymentOptionsSummary.isml

```
<div class="payment-details">
  <isloop items="${pdict.order.billing.payment.selectedPaymentInstruments}" var="payment">
    <isif condition="${payment.paymentMethod === 'CREDIT_CARD'}">
      <isinclude template="checkout/billing/paymentOptions/creditCardSummary" />
    </isif>
    <isif condition="${payment.paymentMethod === 'Radom'}">
      <isinclude template="radom/checkout/radomPaymentSummary" />
    </isif>
  </isloop>
</div>
```

## Step 7

In the metadata folder we have a XML file. Upload the XML file in Administration > Site Development > Import & Export > Upload

## Step 8

Import the uploaded file from Administration > Site Development > Import & Export > Import. It will create a custom object name "radom\_API\_checkout\_data".

## Step 9

Create a data in custom object for API Callout follow path Merchant Tools > Custom Objects > Manage Custom Objects.

- Key: api\_data
- cancelUrl (STRING): Radom-Cancel
- successUrl (STRING): Radom-PlaceOrder
- Customizations (JSON):

```
{
  "leftPanelColor": "linear-gradient(125deg,
    rgba(0,0,128,1) 0%,
    RGBA(196,22,196,1) 100%)",
  "primaryButtonColor": "#000000",
  "slantedEdge": true,
  "allowDiscountCodes": true
}
```

Note: This part is customizable.

- Gateway(JSON):

```
{
  "managed": {
    "methods": [
      {
        "network": "SepoliaTestnet"
      },
      {
        "network": "SepoliaTestnet"
      },
      {
        "network": "PolygonTestnet"
      }
    ]
  }
}
```

**Note:** This is the customizable part.



- Radom API Token (mandatory): Create a account on Radom then create API Token using this link

<https://dashboard.radom.network/developers?tab=API%20tokens>

Add the generated token in Radom API Token.

- Radom webhook verification key (mandatory): Create a link for controller Radom-webhook. To generate the Webhook Verification Key, register your generated link here:

<https://dashboard.radom.network/developers>

- Charge Customer Network Fee: If you want to exclude the radom network platform fee  
uncheck the checkbox.

It should look like this.

Random checkout

Cancel URL

Random-Cancel

Success url:

Random-PlaceOrder

random API checkout data:

api\_data

Customizations:

{  
  "leftPanelColor": "linear-gradient(125deg, rgba(0,0,128,1) 0%, RGBA(196,22,196,1) 100%)",  
  "primaryButtonColor": "#000000",  
  "slantedEdge": true,  
  "allowDiscountCodes": true  
}

Gateway:

{  
  "URL": "https://esdms37f437ef718e2e27/b0d04962c3303092E",  
  "network": "SepoliaTestnet",  
  "network": "PolygonTestnet"  
}

Charge Customer Network Fee:

☐

Random API Token:

eyJhZGRyZXNzIjpuZDhsLCJvcmdhbnRlYXRpb25laWQOUkYTRHY

Random webhook verification key:

YRJZDVnQtNDAA2OS00YWYyLWFoTmMGE0YTSMTBjYtc0ZG

## Step 10

Import the Payment Methods and Payment Processor from Administration > Site Development > Site Import & Export

## Step 11

Import the Web Services from Administration > Operations > Import & Export > services