

## Contents

Installation Guide for Radial's PTF Connector .....	3
Section 1: Install the Radial Module.....	4
Section 2: Configure Radial Core Admin.....	9
Complete Radial Onboarding .....	9
Configure API Details .....	9
Section 3: Configure Payment Methods .....	10
Configure Credit Card Payments .....	11
Set Configurations for Validate Card .....	11
Configure Radial Credit Card.....	13
Configure PayPal Payments .....	16
Configure PayPal Smart Button.....	22
Configure Private Label Credit Card, Ajax Tokenization, and 3DSecure .....	27
Configure General Settings .....	27
Enable Private Label Credit Card Payments .....	28
Enable 3DSecure for Credit Card Payments .....	31
Configure Bank Transfer Payments .....	33
Add Euro to Allowed Currencies.....	33
Configure Radial Bank Transfer .....	33
Enable iDEAL Bank Transfer Payments .....	35
Enable Sofort Bank Transfer Payments .....	35
Configure Alipay Payment.....	37
Complete General Configurations for Alipay .....	37
Enable Alipay Payments.....	37
Configure OpenInvoice Payments .....	39
Add Euro to Allowed Currencies.....	39
Complete General Configurations for OpenInvoice .....	39
Enable OpenInvoice Payments .....	40
Configure Google Pay Payments.....	42
Configure Klarna Payments .....	44

Configure Interac Online Payments .....	46
Add Canadian Dollar to Allowed Currencies.....	46
Enable Interac Online Payments .....	46
Configure Apple Pay Payments .....	48
Configure Link Money Payments .....	49
Section 4: Configure Gift Cards .....	55
Enable the Gift Card Module .....	55
Configure Magento Gift Card Defaults.....	56
Section 5: Configure Tax.....	58
Enable the Tax Module.....	58
Configure Origin Settings for Shipping .....	59
Create Product Tax Classes and Assign to Products .....	59
Select the Tax Class and County of Manufacture for a Product .....	62
Section 6: Configure Radial Fraud.....	64
Configure Risk.....	64
Configure Risk Insight .....	67
Section 7: Configure Address Validation .....	68
Section 8: Configure Webhook .....	69



## Installation Guide for Radial's PTF Connector

Radial provides pre-built connectors to integrate your Adobe Commerce/Magento webstore with Radial's ecommerce offerings. This installation guide walks you through the process of installing and configuring the connector for Radial Payment Solutions, which comprise payment, tax, and fraud (PTF) APIs and services.

Before you can begin detailed integration using the Radial connector, you must contact Radial so that Radial Support can prepare to accept API calls from your webstore. For more information see "Complete Radial Onboarding" in Section 2.

The integration process begins with installing Radial's PTF connector on your Magento server. See "Section 1: Install the Radial Module."

After installation, you must complete the following configurations:

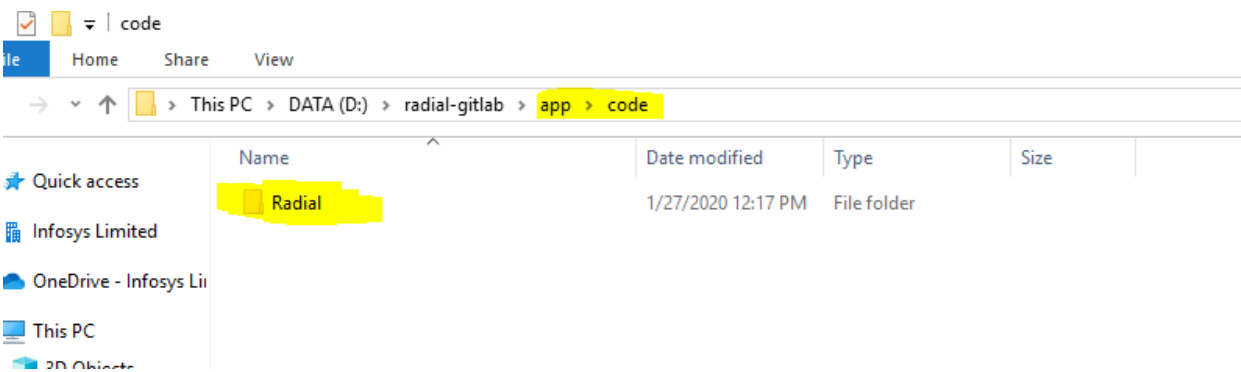
- Radial Core Admin. See Section 2.
- Payment Methods. See Section 3. Radial supports many payment tenders. Complete configuration for each payment method that you plan to use in your webstore.
- Gift Cards. See Section 4.
- Tax. See Section 5.
- Fraud. See Section 6.
- Address Validation. See Section 7.
- Webhook. See Section 8.

## Section 1: Install the Radial Module

To set up the Radial Magento2 Connector, the first major step is to install the Radial module in your Magento instance. After you download the Radial archive file, complete the following steps to install the Radial module.

1. Extract the downloaded archive into the `app/code/` folder of your Magento root directory using a command that follows the format of one of the examples below.
  - Example: `unzip folder.zip`
  - Example: `tar -xvf folder.tgz`

In `app/code` a folder named `Radial` is created (`app/code/Radial`). All Radial modules are inside that `Radial` folder; for example, folders named `Afterpay`, `Alipay`, and `core`.



```
File Edit View Search Terminal Help
[root@lvs02magapp009 code]# ls
Radial
[root@lvs02magapp009 code]# pwd
/opt/lampp/htdocs/radial/app/code
[root@lvs02magapp009 code]# ls -latr
total 4
drwxrwxrwx. 5 root root 4096 Jan 14 10:31 ..
drwxrwxrwx. 3 root root  19 Jan 14 10:31 .
drwxrwxrwx. 8 root root  94 Jan 22 19:05 Radial
[root@lvs02magapp009 code]#
```

2. Execute Magento CLI commands.

In the project root folder on the command line (cmd), run the series of commands listed below. These commands install the Radial module.

- php bin/magento setup:upgrade

```
File Edit View Search Terminal Help
Module 'Magento_CatalogUrlRewriteGraphQL':
Module 'Magento_AynchronousOperations':
Module 'Magento_Usps':
Module 'Magento_ConfigurableImportExport':
Module 'Magento_Cybersource':
Module 'Magento_VaultGraphQL':
Module 'Magento_Version':
Module 'Magento_GoogleTagManager':
Module 'Magento_VisualMerchandiser':
Module 'Magento_Swagger':
Module 'Magento_WebapiAsync':
Module 'Magento_WebapiSecurity':
Module 'Magento_ElasticsearchCatalogPermissions':
Module 'Magento_SalesRuleSampleData':
Module 'Magento_WeeeGraphQL':
Module 'Magento_WeeeStaging':
Module 'Magento_CatalogStagingPageBuilder':
Module 'Magento_WidgetSampleData':
Module 'Magento_GiftCardStaging':
Module 'Magento_WishlistAnalytics':
Module 'Magento_WishlistGraphQL':
Module 'Magento_MultipleWishlistSampleData':
Module 'Magento_Worldpay':
Module 'Amazon_Core':
Module 'Amazon_Login':
Module 'Amazon_Payment':
Module 'Dotdigitalgroup_Email':
Module 'Dotdigitalgroup_Enterprise':
Module 'Klarna_Core':
Module 'Klarna_Ordermanagement':
Module 'Klarna_Kp':
Module 'Magento_PaypalReCaptcha':
Module 'MSP_TwoFactorAuth':
Module 'Radial_Core':
Module 'Radial_CreditCard':
Module 'Radial_PayPal':
Module 'Radial_Payments':
Module 'Radial_Risk':
Module 'Radial_SaveCreditCard':
Module 'Temando_Shipping':
Module 'Vertex_Tax':
Nothing to import.
```

- php bin/magento setup:di:compile

```
File Edit View Search Terminal Help
Mate Terminal
[root@lvs02magapp009 radial]# /opt/lampp/bin/php bin/magento setup:di:compile
Compilation was started.
Interception cache generation... 7/7 [=====] 100% 49 secs 436.0 MiB
Generated code and dependency injection configuration successfully.
[root@lvs02magapp009 radial]#
```

- php bin/magento setup:static-content:deploy -f

```
File Edit View Search Terminal Help
Mate Terminal
[root@lvs02magapp009 radial]# /opt/lampp/bin/php bin/magento setup:static-content:deploy -f
Deploy using quick strategy
frontend/Magento/blank/en_US      2476/2476      ===== 100% 18 secs
adminhtml/Magento/backend/en_US   3026/3026     ===== 100% 26 secs
frontend/Magento/luma/en_US       2493/2493     ===== 100% 28 secs

Execution time: 78.188869953156
[root@lvs02magapp009 radial]#
```

- php bin/magento cache:flush

```
mate terminal
File Edit View Search Terminal Help
[root@lvs02magapp009 radial]# /opt/lampp/bin/php bin/magento cache:flush
Flushed cache types:
config
layout
block_html
collections
reflection
db_ddl
compiled_config
eav
customer_notification
config_integration
config_integration_api
target_rule
full_page
config_webservice
translate
vertex
[root@lvs02magapp009 radial]#
```

- php bin/magento cache:clean

```
Mate Terminal
File Edit View Search Terminal Help
[root@lvs02magapp009 radial]# /opt/lampp/bin/php bin/magento cache:clean
Cleaned cache types:
config
layout
block_html
collections
reflection
db_ddl
compiled_config
eav
customer_notification
config_integration
config_integration_api
target_rule
full_page
config_webservice
translate
vertex
[root@lvs02magapp009 radial]#
```

- chmod -R 777 generated/ pub/static var/ var/log var/cache

```
Mate Terminal
File Edit View Search Terminal Help
[root@lvs02magapp009 radial]# chmod -R 777 var/ var/cache/ var/log generated/
[root@lvs02magapp009 radial]#
```

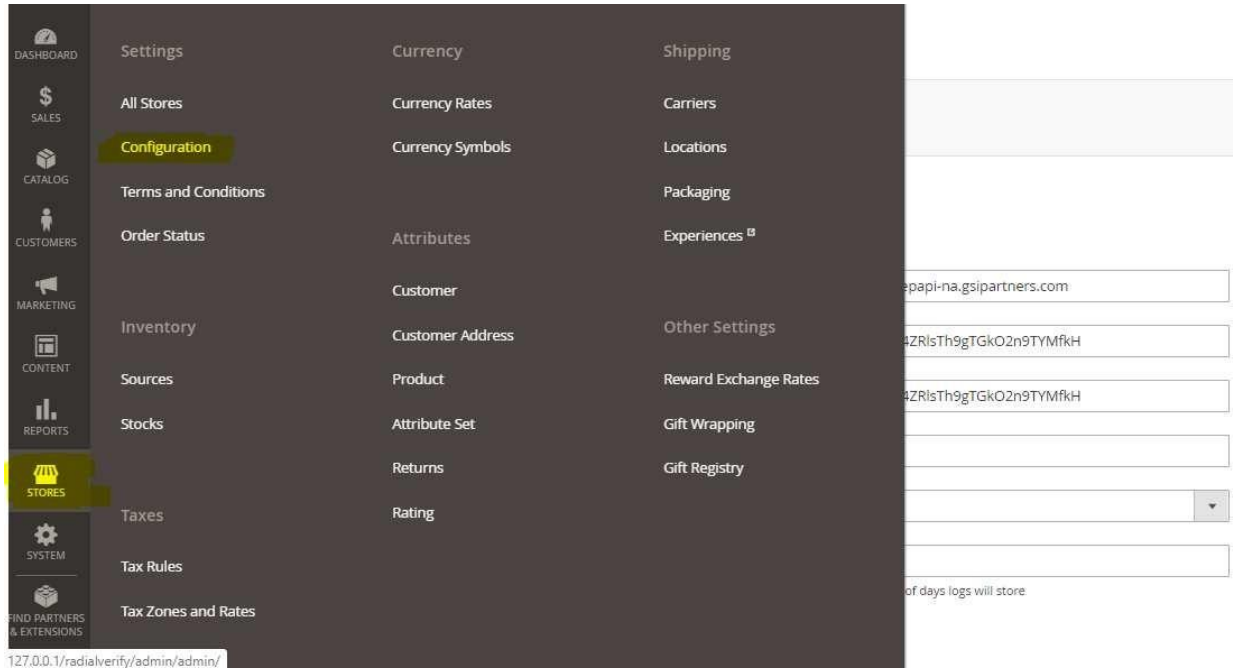
3. Install the Magento crons, by running the following command:
























```
bin/magento cron:install
```

The Radial module is now installed. You can see the **Radial** tab in the Admin section by navigating to **Stores >**

## Configuration.

The following pictures show the navigation and the newly installed Radial module.



-  Dashboard
-  Sales
-  Catalog
-  Customers
-  Marketing
-  Content
-  Radial
-  ROM Settings
-  Reports
-  Stores
-  System
-  Find Partners & Extensions
-  Dashboard
-  Sales
-  Catalog
-  Customers
-  Marketing
-  Content
-  Radial
-  ROM Settings
-  Revenue Catalog
-  Customers
-  Marketing
-  Content
-  Radial
-  ROM Settings

## Configuration

Scope: Default Config ?

- RADIAL** ^
- API Details**
- Address Validation
- Save CreditCard
- Gift Card
- Tax (TDF)
- Risk
- RADIAL ROM** ∨
- GENERAL** ∨
- CATALOG** ∨
- SECURITY** ∨
- CUSTOMERS** ∨
- SALES** ∨
- SERVICES** ∨
- ADVANCED** ∨

### General Configuration

API Url [store view]

Sandbox API key [store view]

To get API Url, API key, and STORE ID. Register first on the given link [here](#).

Store ID [store view]

Environment [store view]

Enable Refund For Precapture Tenders If Order Is Cancelled [store view]

Select

- Radial Afterpay
- Radial Alipay
- Radial Interac Payment

Select One or More

Stop Auth Cancel For Complete Order [store view]

Select

- Radial Credit Card

Select One or More

Retry Confirm Funds [store view]

Provided number of days to retry confirm funds

Save Log, Days [store view]

Provided number of days logs will store

Enable Order Statuses for Settlement [store view]

- Partially Cancelled
- Partially Fulfilled and Invoiced
- Partially Released
- Partially Return Invoiced
- Partially Return Order Invoiced
- Partially Scheduled
- Processing
- Processing
- Ready to Ship
- Received





## Section 2: Configure Radial Core Admin

### Complete Radial Onboarding

Before you can begin detailed integration using the Radial connector, you must contact Radial so that Radial Support can prepare to accept API calls from your store.

- Radial will add your store's domain name to the list of domain names that are allowed to make JavaScript calls (under CORS) at Radial's servers.
- Radial will assign a unique store code for your store. Radial uses the store code to segment orders.
- Radial will provide API keys specific to your store for secured access to Radial's APIs.
- Give Radial your webhook HTTPS endpoint. Radial will use the endpoint to post notifications in real time.

Please send an email to [DL-PTFSupport@radial.com](mailto:DL-PTFSupport@radial.com) expressing your interest and Radial will work with you to onboard in a sandbox environment.

### Configure API Details

Navigate to the API Details tab and complete the following steps in the General Configuration section.

1. In **Environment**, make a selection from the dropdown. The choices are *Sandbox* or *Production*.
2. In **API url**, enter the URL for your store's Radial API calls. Radial will provide this value.
3. In **Sandbox API key** and **Production API key**, enter the API key text strings provided by Radial for your store.
4. In **Store ID**, enter the unique Store ID value provided by Radial for your store.
5. In **Enable refund for pre-capture tenders if order is cancelled**, select the tenders to enable for refund if an order is canceled by fraud or through a manual order cancel from an admin.
6. In **Stop auth cancel for complete order**, select the tenders to stop the cancel of a payment authorization if an order is canceled by fraud or through a manual order cancel from admin.
7. In **Retry confirm funds**, enter the number of days to retry confirm funds if the initial attempt to confirm funds fails.
8. In **Enable order statuses for Settlement**, select the order status values for which shipment is allowed.
9. In **Save Log, Days**, enter the number of days to retain log files. For example, *30*.

## Section 3: Configure Payment Methods

This section describes the various payment methods and tenders supported by the Radial connector. Some of these are region-specific or country-specific and might not apply to you. Please contact your Radial representative to check if a specific payment method or tender is available in your location.

Note that Radial must complete some backend setup and configuration for each payment method that your store uses. Radial handles this process as part of onboarding your store. If you want to add a payment method after the initial onboarding, please contact your Radial representative.

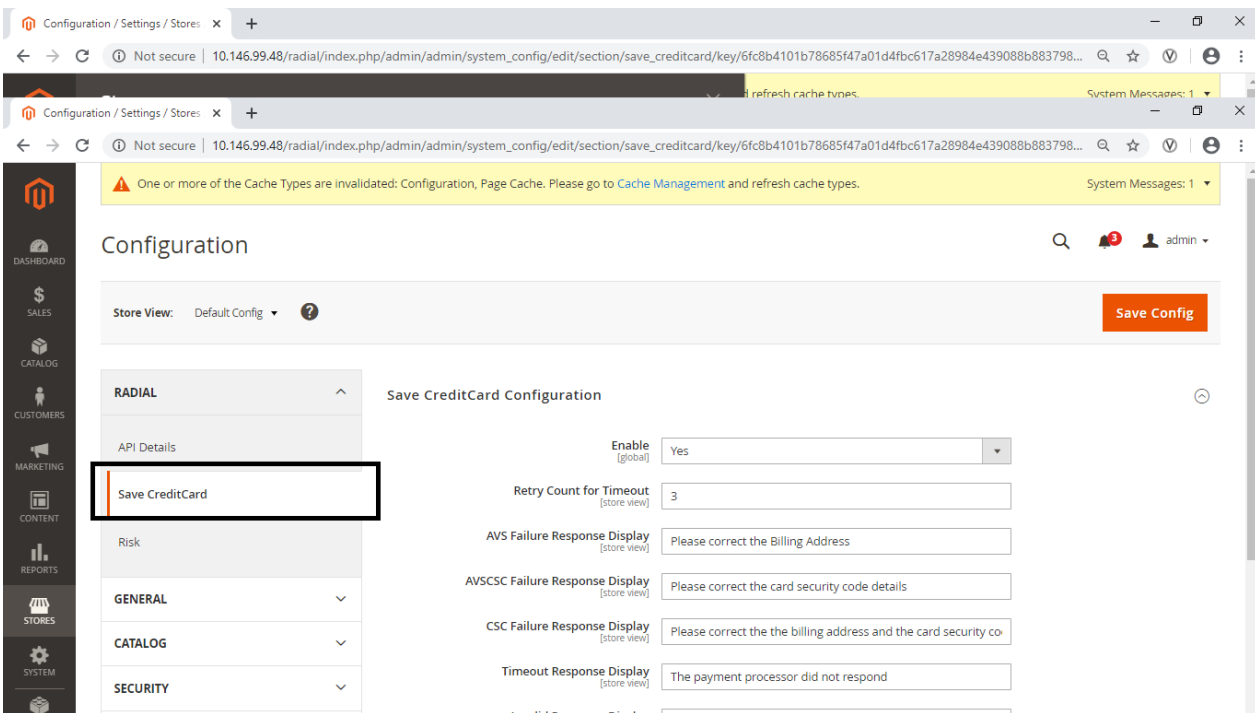
The Connector supports different types of tenders in the following categories:

- Variable Tenders
- Fixed Tenders
- Gifting
- Bank Transfers
- Instant Finance
- Wallets

# Configure Credit Card Payments

## Set Configurations for Validate Card

1. Log on to Magento Admin and navigate to **Stores > Settings > Configuration**.
2. Navigate to **Radial > Save Credit Card**

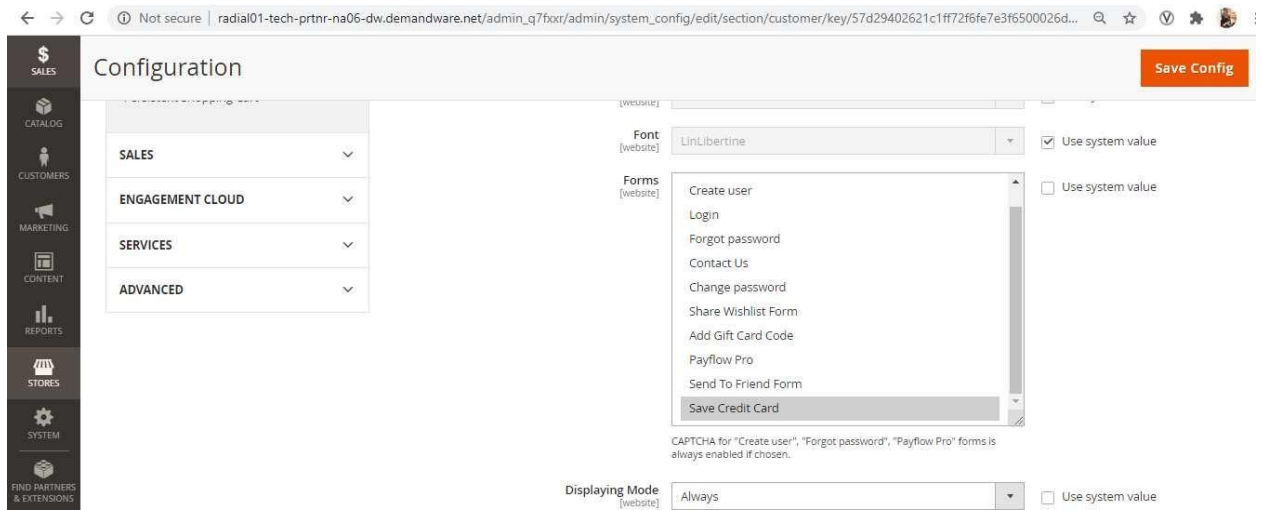


3. In **Enable**, select **Yes** to enable the module to display the API in the customer account frontend. Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.
  - **Retry Count for Timeout:** When the Validate Card API returns a Timeout response, this many retries will be made internally.  
Value: 3
  - **AVS Failure Response Display:** This text is displayed to the customer when the Validate Card API returns an AVS response.  
Value: *Please correct the Billing Address*
  - **AVSCSC Failure Response Display:** This text is displayed to the customer when the Validate Card API returns an AVSCSC response.  
Value: *Please correct the card security code details*
  - **CSC Failure Response Display:** This text is displayed to the customer when the Validate

Card API returns a CSC response.

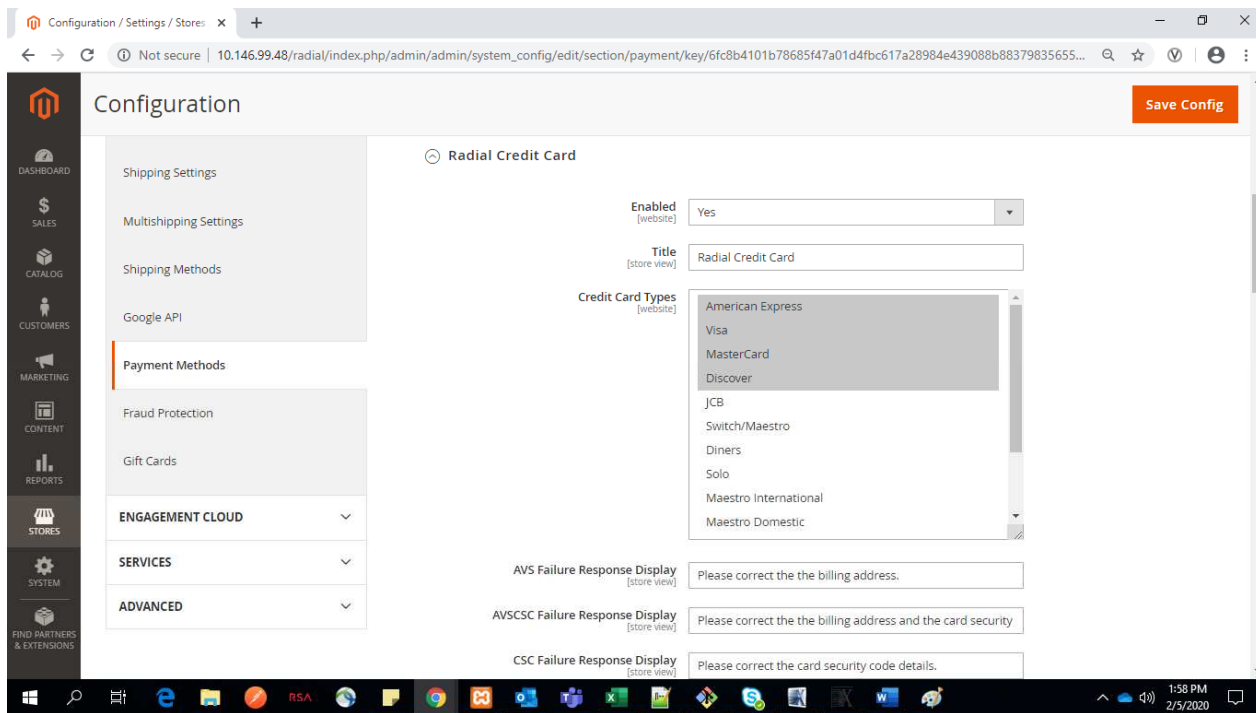
Value: *Please correct the billing address and the card security code*

- **Timeout Response Display:** This text is displayed to the customer when the Validate Card API returns a *Timeout* response.  
Value: *The payment processor did not respond*
- **Invalid Response Display:** This text is displayed to the customer when the Validate Card API returns an *Invalid* Response.  
Value: *The credit card details are invalid*
- **Error Response Display:** This text is displayed to the customer when the Validate Card API returns an *Error* Response.  
Value: *An internal error occurred*
- **Save Log, Days:** The number of days to retain log files.  
Value: 30
- **Message displayed to customer on save credit card disabled:** Save Credit Card feature has been disabled due to exceeded number of **incorrect** attempts. Please contact customer care for further assistance.
- **Retry Count for Invalid Request:** This many retries are allowed for a request that is rejected as invalid.  
Value: 3
- Click **Save Config**.
- Navigate to **Store > Configuration- > Customer- > Customer Configuration- > CAPTCHA** section.
- In **Forms**, include the Save Credit Card form to enable CAPTCHA for the form.



## Configure Radial Credit Card

1. Log on as a Magento Admin. and navigate to **Stores > Settings > Configuration**.
2. In the left panel, expand **Sales** section and choose **Payment Methods Select Sales > Payment Methods > Radial Credit Card**.

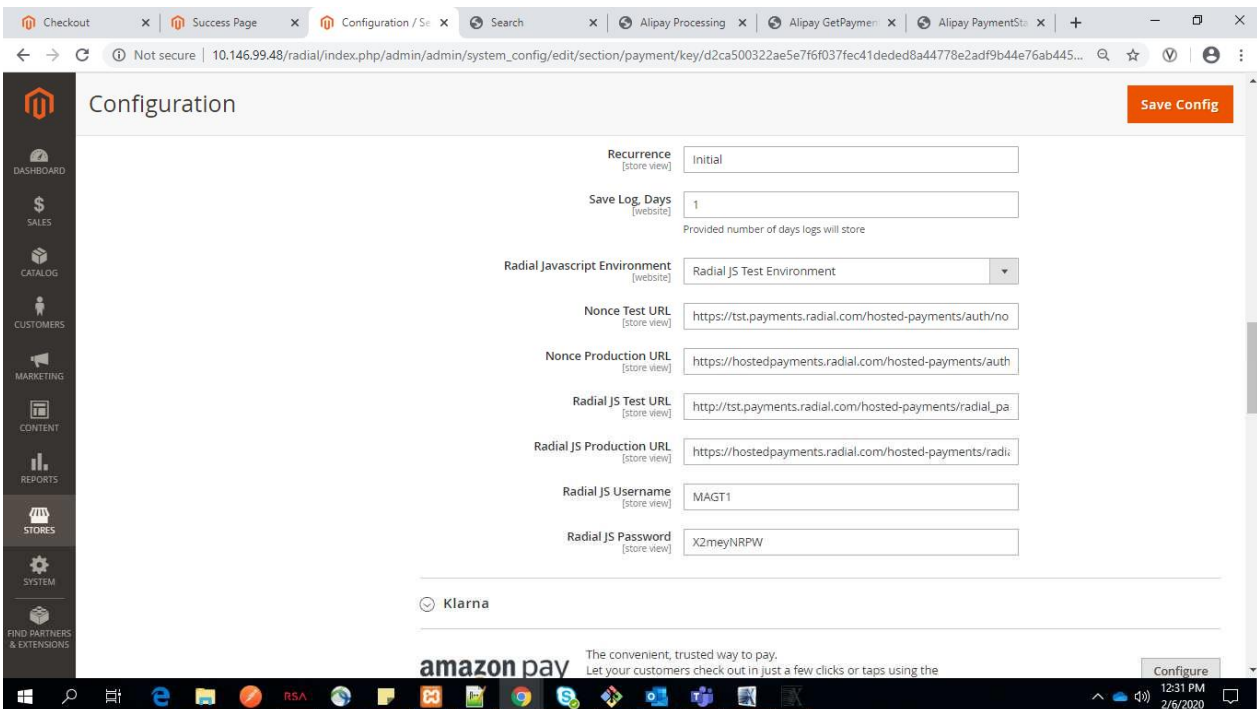


3. In **Enable**, select **Yes** to enable the module to display the Radial Credit Card option of Payment in Customer Checkout.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- In **Title**, enter the title of the Radial Credit Card Payment Flow  
Recommended Value: *Radial Credit Card*
- In Credit Card Types, Select the credit card types to be display as payment options in checkout  
Example value: *Visa, MasterCard, AMEX, Discover*
- **AVS Failure Response Display:** This text is displayed to the customer when the Credit Card Auth returns an AVS Failure Response.  
Value: *Please correct the billing address*
- **AVSCSC Failure Response Display:** This text is displayed to the customer when the Credit Card Auth returns an AVSCSC Failure response.  
Value: *Please correct the billing address and the card security code*
- **CSC Failure Response Display:** This text is displayed to the customer when the Credit Card Auth returns a CSC Failure response.  
Value: *Please correct the card security code*
- **DECLF Failure Response Display:** This text is displayed to the customer when the Credit Card Auth returns an DECLF Failure response.  
Value: *The authorization was declined due to lack of funds or credit*
- **DECLR Failure Response Display:** This text is displayed to the customer when the Credit Card Auth returns an DECLR Failure response.  
Value: *The authorization was declined because the card was reported lost or stolen*
- **PaymentProcessor Timeout Response Display:** This text is displayed to the customer when the Credit Card Auth returns a Timeout response.  
Value: *The payment processor timed out*
- **Configure Number of re-attempts during timeout retry:** This many internal re-attempts must be made when a request returns a a Timeout response.  
Value: *1*
- **Enable AVS/CSC/AVSCSC Retry:** Choose whether to make retry attempts in the case of AVS, CSC, or AVSCSC responses.
- **Configure Number of re-attempts during AVS/CSC/AVSCSC Retry:** This many re-attempts must be made in the case of AVS, CSC, or AVSCSC responses.  
Value: *0*
- **Schema Version:** *1.2*
- **Recurrence:** *Initial*
- **Save Log Days:** The number of days to retain log files.  
Value: *1*

- **Radial JavaScript Environment:** Choose the environment, either *Test* or *Production*
- Radial 3DS Js Origin URL : Need to provide website URL.  
Reference: <https://apay01-mag02.gsipartners.com>
- **Nonce Test URL:** The Nonce URL for the test environment.  
*https://tst.payments.radial.com/hosted-payments/auth/nonce*
- **Nonce Production URL:** The Nonce URL for the Production environment.  
*https://hostedpayments.radial.com/hosted-payments/auth/nonce*
- **Radial JS Test URL:** The Radial JavaScript URL for the test environment.  
*https://tst.payments.radial.com/hosted-payments/v2.0/radial\_payments.js*
- **Radial JS Production URL:** The Radial JavaScript URL for the production environment.  
*https://hostedpayments.radial.com/hosted-payments/radial\_payments.js*
- **Radial JS Username:** The username provided by Radial to use for authentication in JavaScript calls.  
Example: *MAGT2US*
- **Radial JS Password:** password provided by Radial to use for authentication in JavaScript calls.  
Example: *XXXXXXXXX*



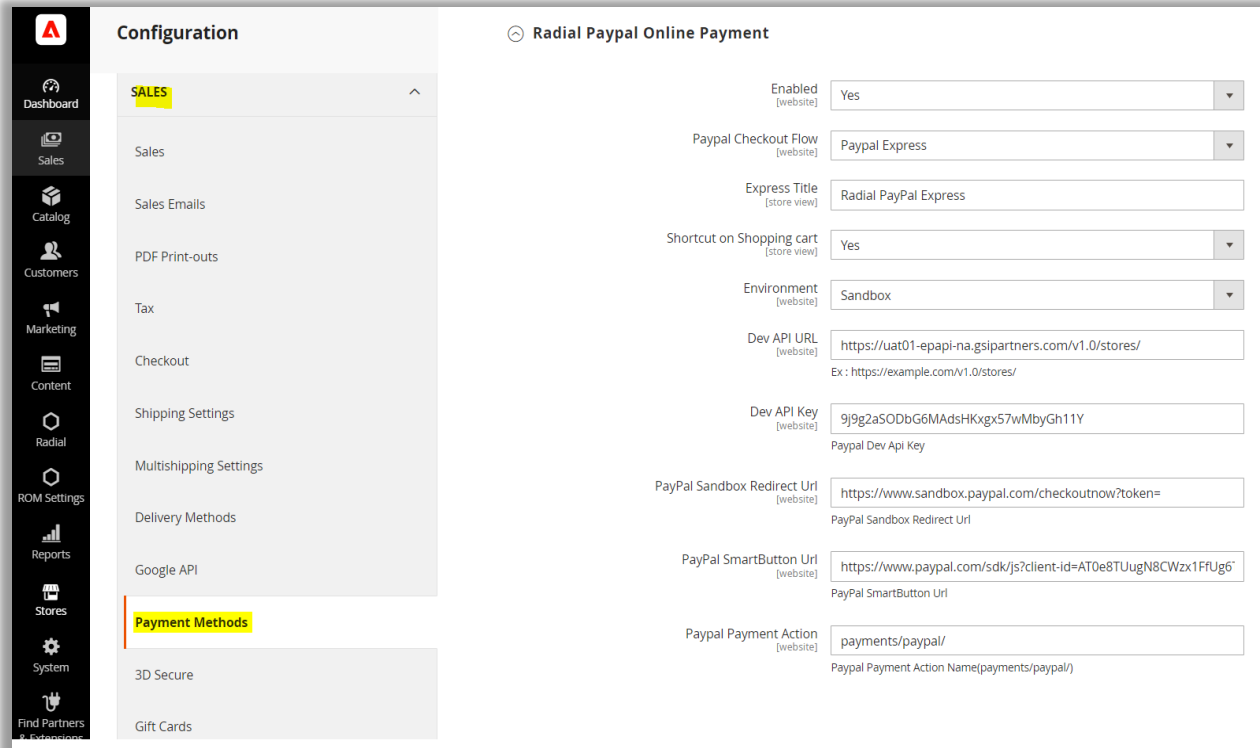
The screenshot shows the 'Configuration' page for the Radial system. The browser address bar indicates the URL is `10.146.99.48/radial/index.php/admin/admin/system_config/edit/section/payment/key/d2ca500322ae5e7f6f037fec41deded8a44778e2ad9b44e76ab445...`. The page features a sidebar with navigation options: DASHBOARD, SALES, CATALOG, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, SYSTEM, and FIND PARTNERS & EXTENSIONS. The main content area contains the following configuration fields:

- Recurrence** (store view): Initial
- Save Log. Days** (website): 1  
Provided number of days logs will store
- Radial Javascript Environment** (website): Radial JS Test Environment
- Nonce Test URL** (store view): `https://tst.payments.radial.com/hosted-payments/auth/no`
- Nonce Production URL** (store view): `https://hostedpayments.radial.com/hosted-payments/auth`
- Radial JS Test URL** (store view): `http://tst.payments.radial.com/hosted-payments/radial_pa`
- Radial JS Production URL** (store view): `https://hostedpayments.radial.com/hosted-payments/radi`
- Radial JS Username** (store view): MAGT1
- Radial JS Password** (store view): X2meyNRPW

At the bottom of the configuration area, there is a section for 'Klarna' and an 'amazon pay' banner with the text: 'The convenient, trusted way to pay. Let your customers check out in just a few clicks or taps using the'. A 'Configure' button is located at the bottom right of the banner. The system tray at the bottom of the screen shows the time as 12:31 PM on 2/6/2020.

## Configure PayPal Payments

1. Log on as a Magento Admin and navigate to **Stores > Settings > Configuration**.
2. In the left panel, expand the **Sales** section and choose **Payment Methods**.
3. Navigate to **Sales > Payment Methods > Radial PayPal Online Payment**.

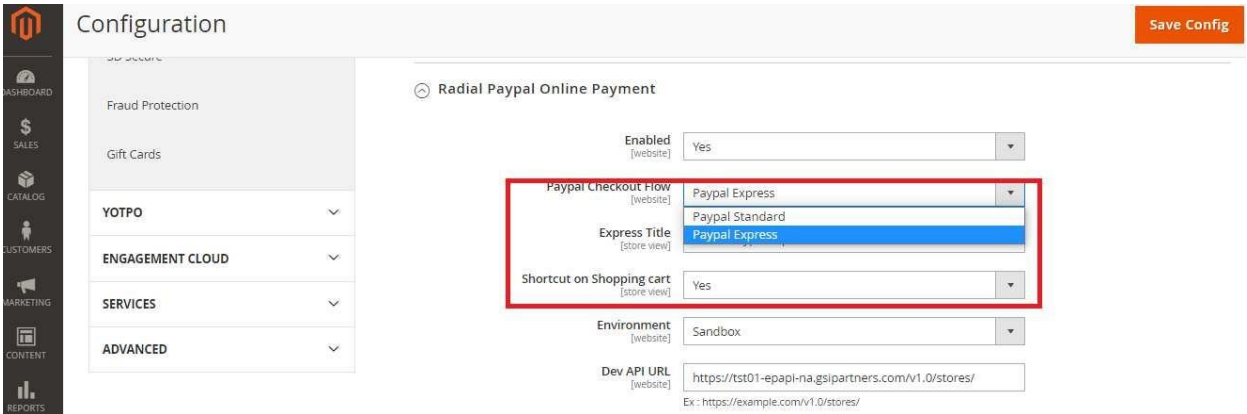


The screenshot shows the Magento Admin Configuration interface. On the left, the 'Configuration' sidebar is expanded to 'SALES' > 'Payment Methods'. The main content area displays the 'Radial Paypal Online Payment' configuration page. The settings are as follows:

Field	Value
Enabled (website)	Yes
Paypal Checkout Flow (website)	Paypal Express
Express Title (store view)	Radial PayPal Express
Shortcut on Shopping cart (store view)	Yes
Environment (website)	Sandbox
Dev API URL (website)	https://uat01-epapi-na.gsipartners.com/v1.0/stores/ Ex: https://example.com/v1.0/stores/
Dev API Key (website)	9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y Paypal Dev Api Key
PayPal Sandbox Redirect Url (website)	https://www.sandbox.paypal.com/checkoutnow?token= PayPal Sandbox Redirect Url
PayPal SmartButton Url (website)	https://www.paypal.com/sdk/js?client-id=AT0e8TUugN8CWzx1FfUg6* PayPal SmartButton Url
Paypal Payment Action (website)	payments/paypal/ Paypal Payment Action Name(payments/paypal/)

4. In **PayPal Checkout Flow**, make a selection. PayPal offers two types of checkout:
  - PayPal Standard
  - PayPal Express





If you select **PayPal Express** as the checkout flow, you can enable the shortcut on the shopping cart button.

If you select **PayPal Standard** as the checkout flow, the shortcut on shopping cart button will not be available.

5. To display the Checkout with PayPal button on the Cart page, set **Shortcut on Shopping cart** to **Yes**.
6. In Environment, select *Sandbox* or *Production*.

If you are using credentials from your sandbox account, select *Sandbox*.

7. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection:

- **API URL** example: *https://example.com/v1.0/stores/*
- **API Key** example: *XXXXXXXXXXXXXXXXXXXXXXXXXXXX*
- **Smart Button URL** example: *https://www.paypal.com/sdk/js?client-id=XXXXXXXXXXXXXXXXXXXXXXXXXXXX &intent=order&enable-funding=venmo&commit=false*
- **PayPal Redirect Url:**
  - *Sandbox* example: *https://www.sandbox.PayPal.com/checkoutnow?token=*
  - *Production* example: *https://www.PayPal.com/checkoutnow?token=*

⊖ Radial Paypal Online Payment

Enabled [website] Yes

Paypal Checkout Flow [website] **Paypal Express**

Express Title [store view] **Radial PayPal Express**

Shortcut on Shopping cart [store view] Yes

Environment [website] Sandbox

Dev API URL [website]   
Ex : https://example.com/v1.0/stores/

Dev API Key [website]   
Paypal Dev Api Key

⊖ Radial Paypal Online Payment

Enabled [website] Yes

Paypal Checkout Flow [website] **Paypal Standard**

Title [store view] Radial Paypal

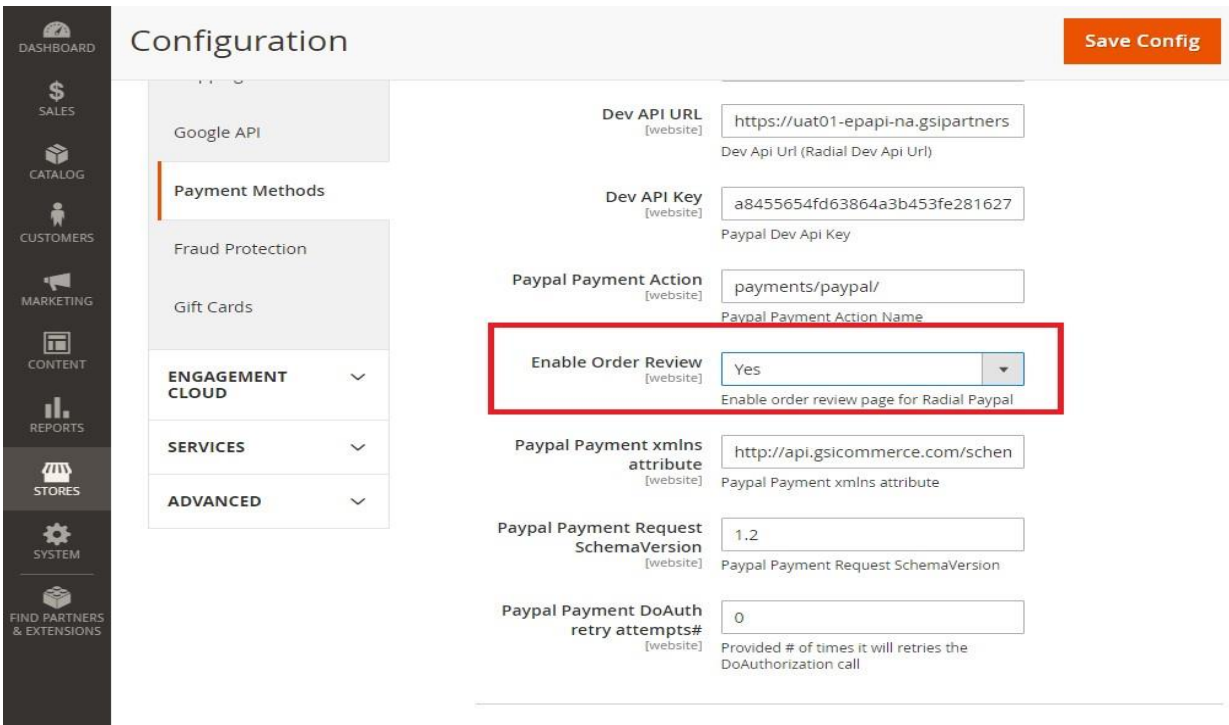
Environment [website] Sandbox

Dev API URL [website]   
Ex : https://example.com/v1.0/stores/

Dev API Key [website]   
Paypal Dev Api Key

PayPal Sandbox Redirect Url [website]   
PayPal Sandbox Redirect Url

8. Enter a **Title** to identify the selected checkout flow as a payment method during checkout.



9. To display the Order Review Page, set **Enable Order Review** to Yes.



10. **PayPal Payment xmlns attribute** is a Radial API request attribute.

Recommended value: *http://api.gsicommerce.com/schema/checkout/1.0*

11. **PayPal Payment Request Schema Version** is the schema version for the Radial API request

Recommended value: *1.2*

12. **PayPal Payment DoAuth retry attempts#**: When the DoAuth API returns a Timeout response, this many retries will be made internally.

Recommended value: *3*

13. **PayPal NoShippingAddressDisplay**: The default option should be 2. The choices are 0, 1, and 2. Refer to the screenshot below for an explanation of the three options and their meanings.

14. **Save Log, Days**: The number of days to retain log files.

15. **ALLOW PAYPAL UNVERIFIED BUYER**: Select Yes to allow unverified buyers or No to allow only buyers that have been verified by PayPal.

16. **Default Phone Number**: If the phone number is missing from PayPal payer's account, the default phone number entered here is used. This allows for smoother order flow.



17. **PayPal payer phone number missing message:** Message displays to customer if phone number is missing in their PayPal account.

The image below shows the default configuration options.

Configuration

Radial Paypal Online Payment

Enabled

Paypal Checkout Flow

Express Title

Shortcut on Shopping cart

Environment

Dev API URL   
Ex: https://example.com/v1.0/stores/

Dev API Key   
Paypal Dev Api Key

PayPal Sandbox Redirect Url   
PayPal Sandbox Redirect Url

PayPal SmartButton Url   
PayPal SmartButton Url

Paypal Payment Action   
Paypal Payment Action Name(payments/paypal/)

Enable Order Review   
Enable order review page for Radial Paypal

Paypal Payment xmlns attribute   
Paypal Payment xmlns attribute(http://api.gsicommerce.com/schema/checkout/1.0)

Paypal Payment Request SchemaVersion   
Paypal Payment Request SchemaVersion(1.2)

Paypal Payment DoAuth retry attempts#   
Provided # of times it will retries the DoAuthorization call

Paypal NoShippingAddressDisplay   
0: Shipping address is displayed in PayPal review page.  
1: Shipping address is not displayed in PayPal review page(Not applicable if customer place order from Shopping cart page by using PayPal Shortcut button).  
2: Shipping address is displayed in PayPal review page. If you do not pass the shipping address, PayPal obtains it from the buyer's account profile.

Save Log, Days   
Provided number of days logs will store

PayPal Smart Button on Shopping cart

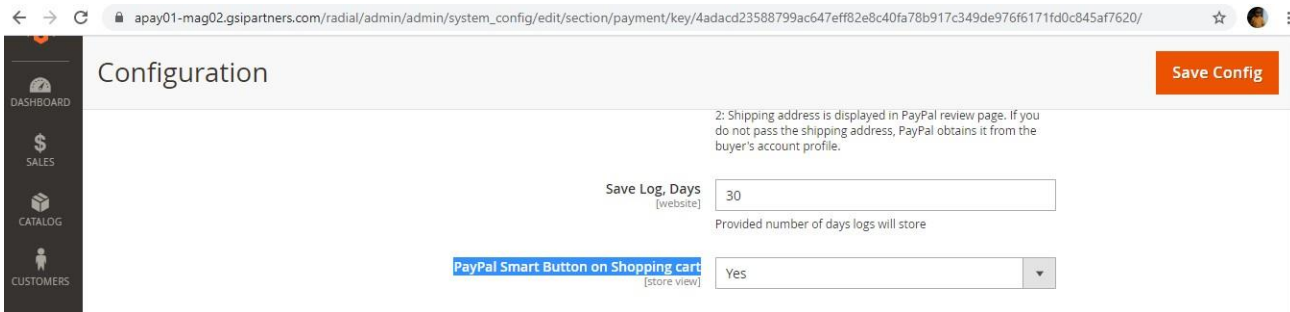
ALLOW PAYPAL UNVERIFIED BUYER

Default Phone Number   
Default phone number , if paypal payer phone field is empty

Paypal payer phone number missing message   
Message to display if phone number is missing from payers paypal account

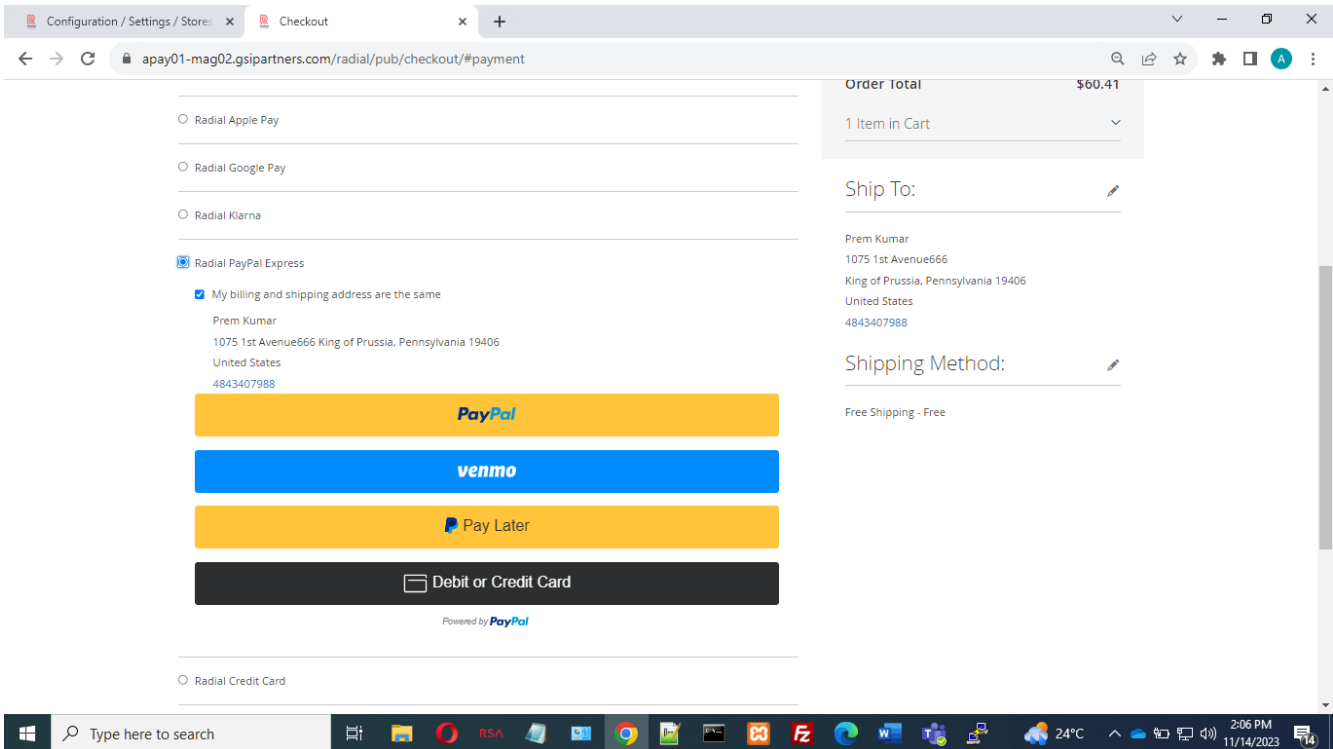
## Configure PayPal Smart Button

1. Log on as a Magento Admin and navigate to **Stores > Settings > Configuration**.
2. In the left panel, expand **Sales** section and choose **Payment Methods Select Sales > Payment Methods > Radial PayPal Online Payment**.
3. **PayPal Smart Button on Shopping cart**: select Yes to display the PayPal Smart Button.

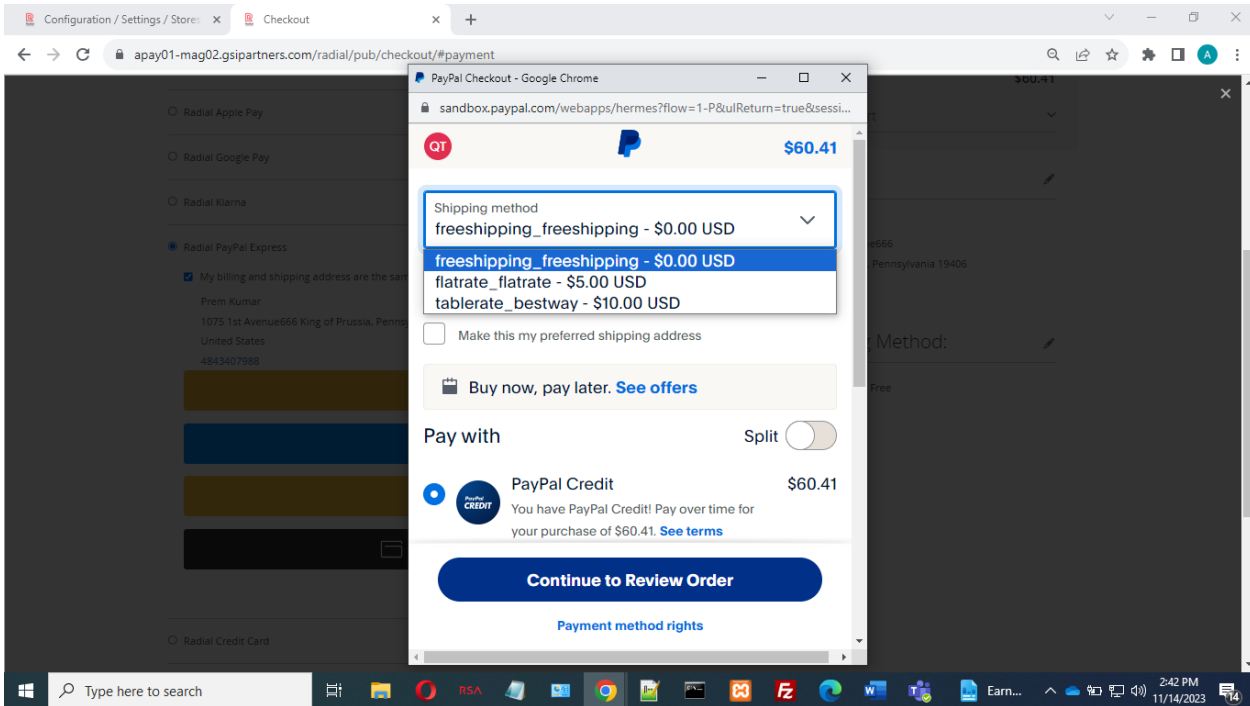


## Radial PayPal V2 Payment frontend Experience:

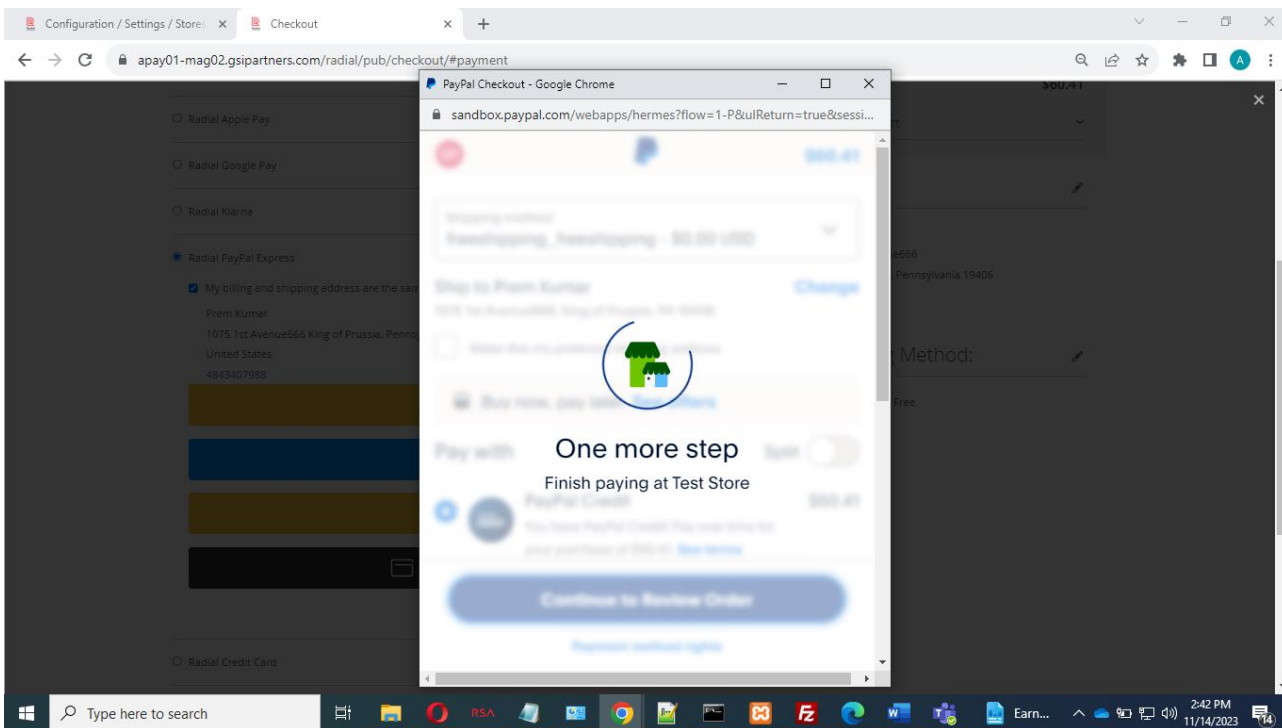
### Paypal Section in Checkout Page:



After click PayPal button in checkout page it will redirect to PayPal payment gateway.



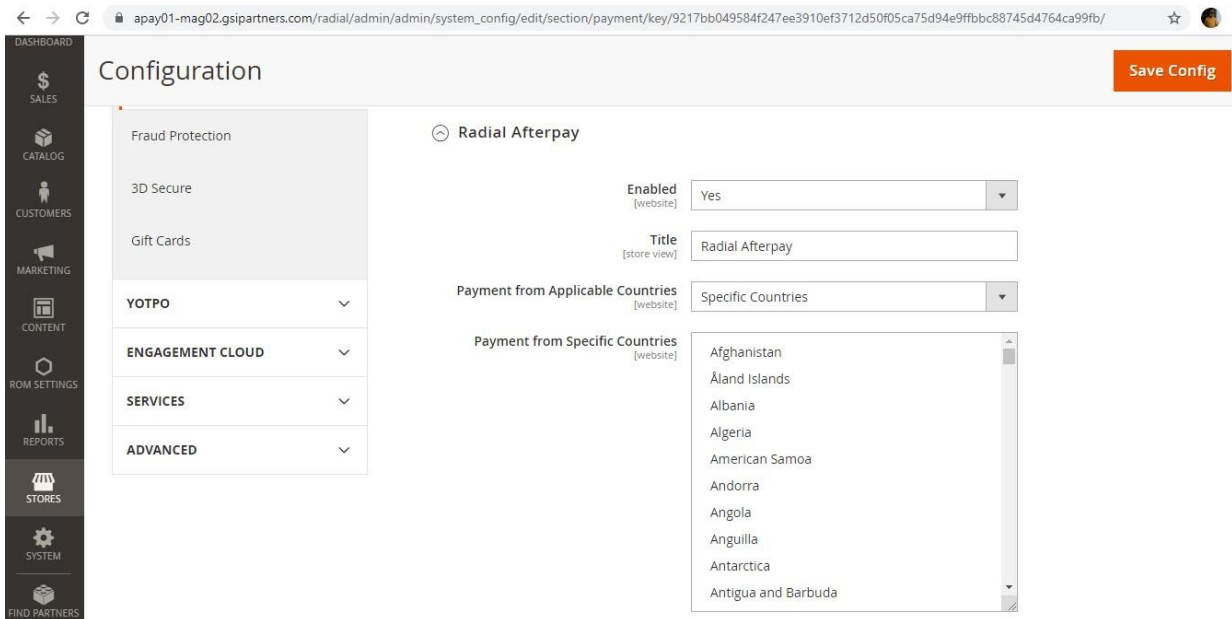
After payment confirmation from the PayPal it will redirect to success page.



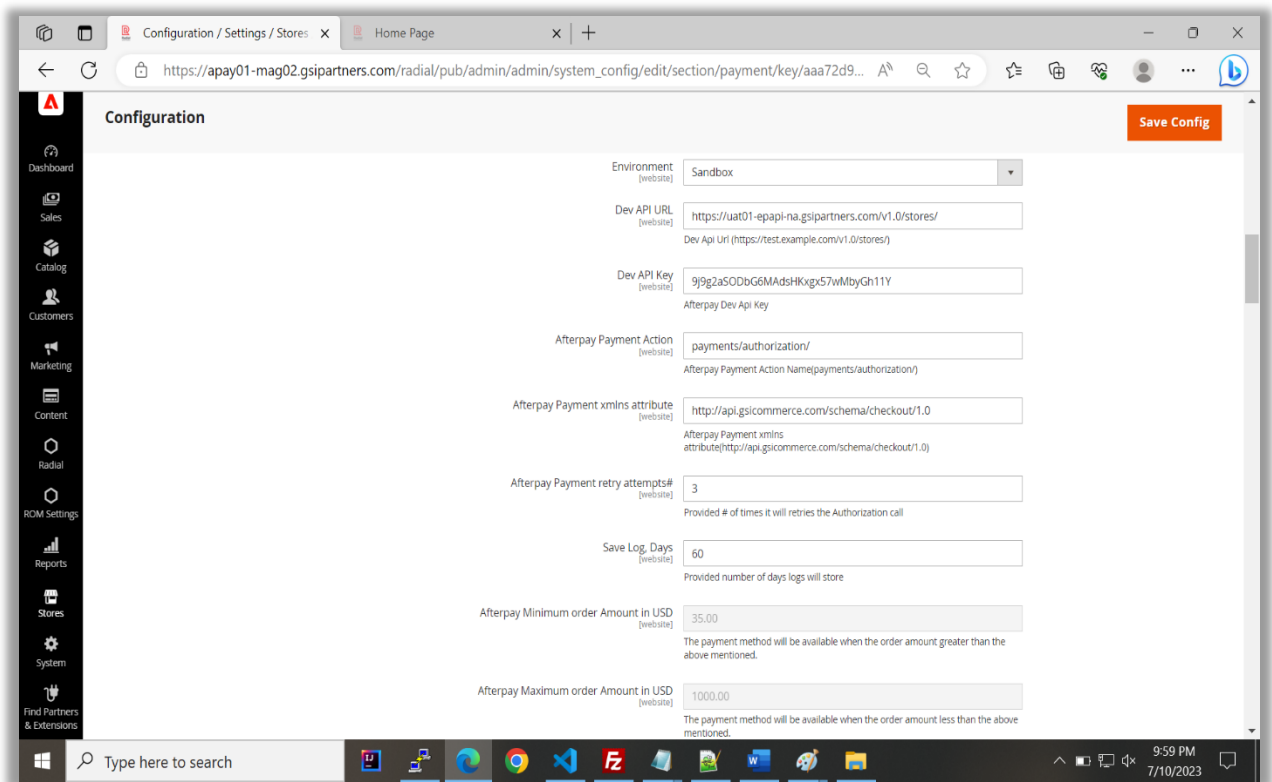
## Configure Afterpay Payments

**Restriction:** This payment method can be used only in the US and other specifically supported countries.

1. Log on as a Magento Admin and navigate to **Stores > Settings > Configuration**.
2. In the left panel, expand **Sales** section and choose **Payment Methods**. Select **Sales > Payment Methods > Radial Afterpay**.
3. In **Title**, enter a text label to identify this payment method during checkout.
4. In Environment, select *Sandbox* or *Production*.  
If you are using credentials from your sandbox account, select *Sandbox*.
5. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection:
  - API URL example: *https://example.com/v1.0/stotes/*
  - API Key example: *XXXXXXXXXXXXXXXXXXXXXXXXXXXX*

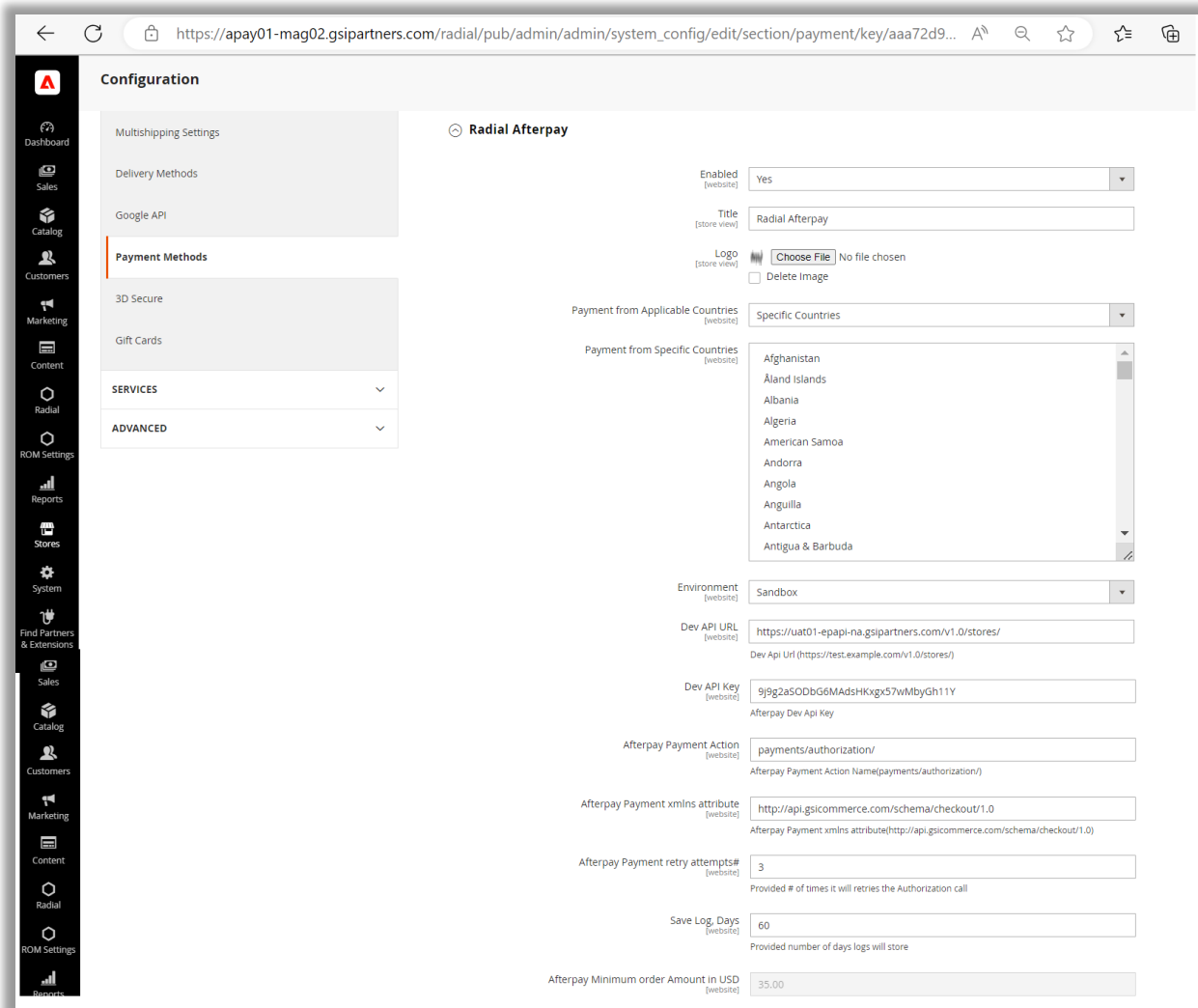






- **Afterpay Payment Action:** Value should be *payments/authorization/*
- **Afterpay Payment xmlns attribute:** This is a Radial API request attribute. Value should be: *http://api.gsicommerce.com/schema/checkout/1.0*
- **Afterpay Payment retry attempts#:** When the API returns a Timeout response, this many retries will be attempted internally. Radial suggests a value of 3
- **Save Log, Days:** The number of days to retain log files.
- **Afterpay Minimum order Amount in USD:** The Afterpay payment method will be available only when the order amount is greater than the amount specified here. After you change the value here, it's in disabled mode, and a daily cron job will update the Minimum order amount in the checkout.
- **Afterpay Maximum order Amount in USD:** The payment method will be available when the order amount is less than the amount specified here. After you change the value here, it's in disabled mode, and a daily cron job will update the Maximum order amount in the checkout.

The following screenshot shows default configuration options.



The screenshot shows the 'Radial Afterpay' configuration page in the Radial Admin interface. The page is titled 'Configuration' and has a sidebar on the left with various navigation options. The main content area is divided into two sections: 'Payment Methods' and 'Radial Afterpay'.

**Payment Methods:**

- Multishipping Settings
- Delivery Methods
- Google API
- Payment Methods**
- 3D Secure
- Gift Cards
- SERVICES
- ADVANCED

**Radial Afterpay Configuration:**

- Enabled:** Yes
- Title:** Radial Afterpay
- Logo:** Choose File (No file chosen) / Delete Image
- Payment from Applicable Countries:** Specific Countries
- Payment from Specific Countries:** Afghanistan, Åland Islands, Albania, Algeria, American Samoa, Andorra, Angola, Anguilla, Antarctica, Antigua & Barbuda
- Environment:** Sandbox
- Dev API URL:** https://uat01-epapi-na.gsipartners.com/v1.0/stores/ (Dev Api Uri: https://test.example.com/v1.0/stores/)
- Dev API Key:** sj9g2a5ODbG6MadsHKxgX57wMbyGh11Y (Afterpay Dev Api Key)
- Afterpay Payment Action:** payments/authorization/ (Afterpay Payment Action Name: payments/authorization/)
- Afterpay Payment xmains attribute:** http://api.gsicommerce.com/schema/checkout/1.0 (Afterpay Payment xmains attribute: http://api.gsicommerce.com/schema/checkout/1.0)
- Afterpay Payment retry attempts#:** 3 (Provided # of times it will retry the Authorization call)
- Save Log, Days:** 60 (Provided number of days logs will store)
- Afterpay Minimum order Amount in USD:** 35.00

# Configure Private Label Credit Card, Ajax Tokenization, and 3DSecure

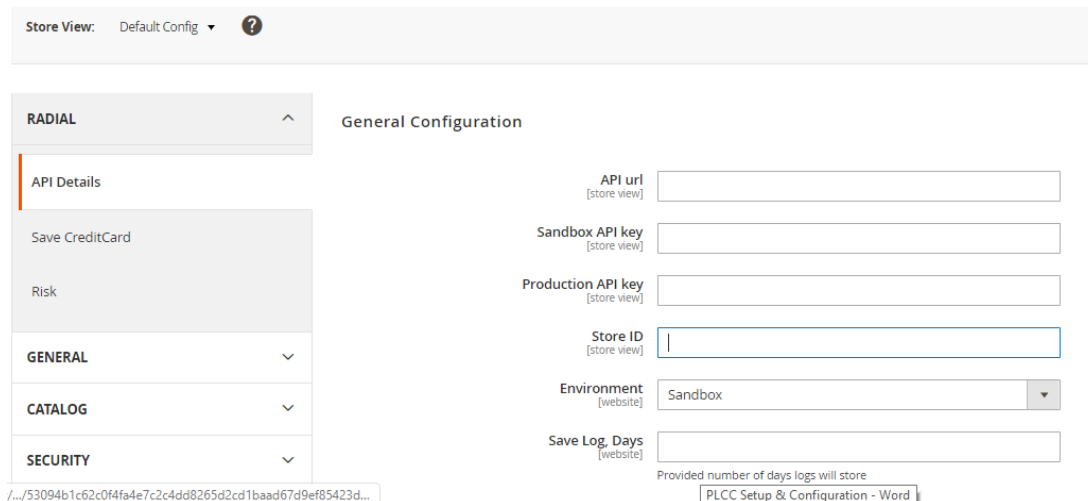
Before you can use functionality that depends on Radial JavaScript, Radial must add your store's domain name to the list of domain names that are allowed to make JavaScript calls to Radial's servers.


The store code, API keys, and URLs are unique to each store. Your values will be different from the ones shown in the examples in the section. Please contact your Radial team for the specific configuration values for your store.


## Configure General Settings

To configure general administrative settings:

1. Navigate to **Store > Configuration > Radial > API Details**.
2. Enter data as follows:
  - In **API url**, enter your store's Radial-provided API URL.  
Example: *https://uat01-epapi-na.gsipartners.com*
  - In **Sandbox API key** and **Production API key**, Enter the API key values provided by Radial.  
Sandbox example: *9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y*  
Production example: *9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y*
  - In **Store ID**, enter the Store ID value provided by Radial  
Example: *MAGT2US*
  - In **Environment**, select *Sandbox* or *Production*.
  - In **Save Log, Days**, enter the number of days to retain log files.  
Example: *30*




Store View: Default Config 


**RADIAL**  General Configuration


API Details

Save CreditCard

Risk

**GENERAL** 

**CATALOG** 

**SECURITY** 

API url [store view]

Sandbox API key [store view]

Production API key [store view]

Store ID [store view]

Environment [website]

Save Log, Days [website]

Provided number of days logs will store

PLCC Setup & Configuration - Word

## Enable Private Label Credit Card Payments

To Enable the Private Label Credit Cards (PLCC) as a Payment Option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Credit Card**.
2. Enter data as follows:
  - In **Radial Credit Card Enabled**, select Yes. This enables Radial Credit Card to display as a payment option in customer checkout.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.
  - In **Title**, enter text to describe the Radial Credit Card Payment Flow.
  - **Radial 3DS Js Origin URL** : Need to provide website URL.  
Reference: <https://apay01-mag02.gsipartners.com>  
Recommended Value: *Radial Credit Card*
  - In **Credit Card Types**, choose *Private Label Credit Card* as one of the credit card types to display in checkout.
  - **AVS Failure Response Display**: This text is displayed to the customer when the Credit Card Auth returns an AVS FailureResponse.  
Value: *Please correct the billing address*
  - **AVSCSC Failure Response Display**: This text is displayed to the customer when the Credit Card Auth returns anAVSCSC Failure Response.  
Value: *Please correct the billing address and the card security code*
  - **CSC Failure Response Display**: This text is displayed to the customer when the Credit Card Auth returns an CSC FailureResponse.  
Value: *Please correct the card security code details*
  - **DECLF Failure Response Display**: This text is displayed to the customer when the Credit Card Auth returns an DECLFFailure Response.  
Value: *The authorization was declined due to lack of funds*
  - **DECLR Failure Response Display**: This text is displayed to the customer when the Credit Card Auth returns a DECLRFailure Response.  
Value: *The authorization was declined because the card has been reported lost or stolen.*
  - **Payment Processor Timeout Response Display**: This text is displayed to the customer when the Credit Card Auth returns a Timeout Response.  
Value: *The payment processor is not responding and Payment auth cancel is acknowledged successfully*
  - **DECL Failure Response Display**: This text is displayed to the customer when the Credit Card Auth

returns an DECLFailure Response.

Value: *The authorization was declined*

- **Configure Number of Re-attempts during Timeout Retry:** This many internal retries will be attempts if Credit Card Auth receives a timeout response.  
Value: *1*
- **Enable AVS/CSC/AVSCSC Retry:** Choose whether to retry in case of an AVS, CSC, or AVSCSC response.
- **Configure Number of Re-attempts during AVS/CSC/AVSCSC Retry:** This many retries will be attempted if there is an AVS, CSC, or AVSCSC response.  
Value: *0*
- **Schema Version:** enter *1.2*
- **Recurrence:** enter *Initial*
- **Save Log Days:** The number of days to retain log files. Value: *1*
- **Radial JavaScript Environment:** Select *Test* or *Production*.
- **Nonce Test URL:** The Nonce URL for the test environment.  
Value: *https://tst.payments.radial.com/hosted-payments/auth/nonce*
- **Nonce Production URL:** The Nonce URL for the production environment.  
Value: *https://hostedpayments.radial.com/hosted-payments/auth/nonce*
- **Radial JS Test URL:** The Radial JavaScript URL for the test environment.  
Value: *https://tst.payments.radial.com/hosted-payments/v2.0/radial\_payments.js*
- **Radial JS Production URL:** The Radial JavaScript URL for the production environment.  
Value: *https://hostedpayments.radial.com/hosted-payments/radial\_payments.js*
- **Radial JS Username:** The username provided by Radial to use for authentication in JavaScript calls.  
Example: *MAGT2US*
- **Radial JS Password:** The password provided by Radial to use for authentication in JavaScript calls.  
Example: *XXXXXXXXX*

## Configuration

- Shipping Settings
- Multishipping Settings
- Shipping Methods
- Google API
- Payment Methods
- Fraud Protection
- Gift Cards
- ENGAGEMENT CLOUD ▼

### Radial Credit Card

**Radial Credit Card Enabled** [website]

**Title** [store view]

**Credit Card Types** [website]

- Switch/Maestro
- Diners
- Solo
- Maestro International
- Maestro Domestic
- Hipercard
- Elo
- Private Label Credit Card
- Aura
- Other

- **PLCC Bin Range:** The Private Label Credit Card bin range.  
Example: *604586*
- **Enable Expiry Date for Private Label Credit Card.** Choose whether to apply expiration dates to Private Label Credit Cards.

[store view]

**PLCC Bin Range** [store view]

The Private Label Credit Card Bin Ranging : If more than one, mention as comma separated values.Eg:604586,4578

**Enable Expiry Date For Private Label Credit Card** [website]

**Purge Credit Card Table** [store view]

Enter in Number of days

3. Save the Configuration.

## Enable 3DSecure for Credit Card Payments

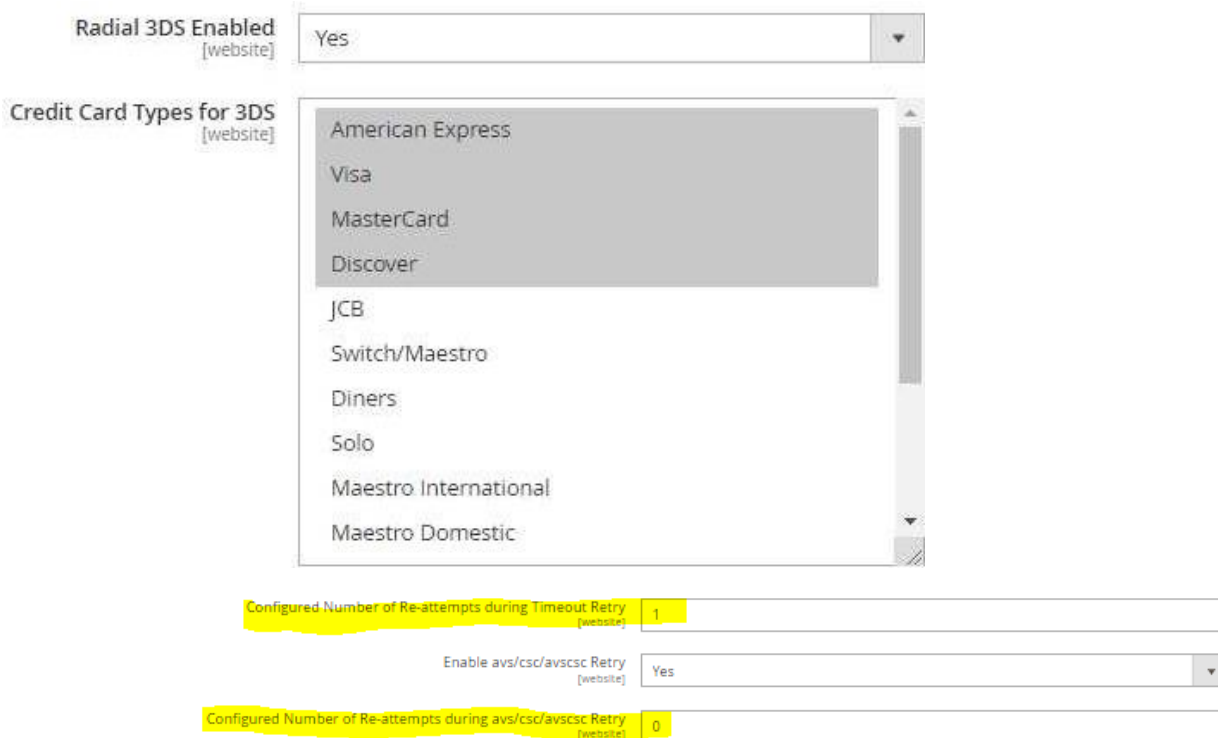
To enable 3DSecure (3DS) for Radial Credit Card payments:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Credit Card**.
2. In **Radial 3DS Enabled**, select Yes.
3. In **Credit Card Types for 3DS**, choose one or more credit card types to enable.
4. Save the configuration.
5. Perform a cache flush.
6. **Configure Number of Re-attempts during Timeout Retry:** This many internal retries will be attempted if Credit Card Auth receives a timeout response.

Value: 1

7. **Configure Number of Re-attempts during AVS/CSC/AVSCSC Retry:** This many retries will be attempted if there is an AVS, CSC, or AVSCSC response.

Value: 0

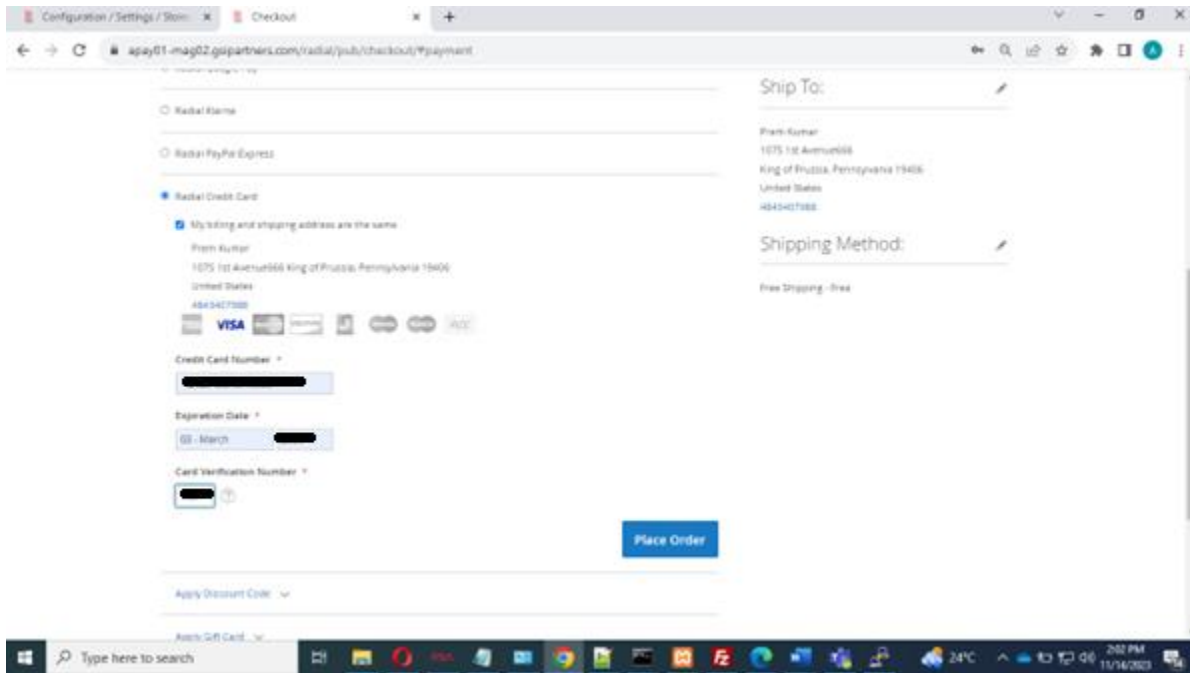


The screenshot shows the configuration interface for Radial 3D Secure. It includes the following elements:

- Radial 3DS Enabled [website]:** A dropdown menu set to "Yes".
- Credit Card Types for 3DS [website]:** A list of credit card types with checkboxes, including American Express, Visa, MasterCard, Discover, JCB, Switch/Maestro, Diners, Solo, Maestro International, and Maestro Domestic.
- Configured Number of Re-attempts during Timeout Retry [website]:** A text input field containing the value "1".
- Enable avs/csc/avscsc Retry [website]:** A dropdown menu set to "Yes".
- Configured Number of Re-attempts during avs/csc/avscsc Retry [website]:** A text input field containing the value "0".

Credit Card 3DS V2.0 Checkout Experience:

Reference screen shot for credit card form,



Once we enter credit card details and if its 3DS enabled it will show the 3DS pop-up window.



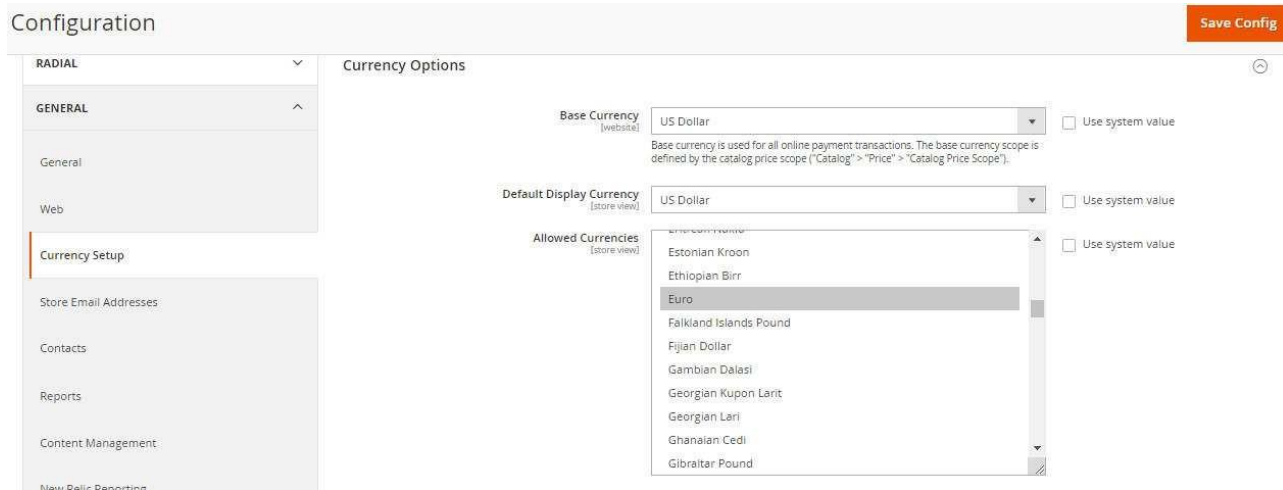
## Configure Bank Transfer Payments

The Bank Transfer payment method can be used only in specific countries where it is supported, such as German and The Netherlands. The store should be set to allow Euro as a currency, with US Dollar as base currency.

### Add Euro to Allowed Currencies

There is a global configuration, Allowed Currencies, where you can select the allowed currencies for the store.

1. Navigate to **Store > Configuration > General > Currency Setup**.
2. In **Base Currency**, select *US Dollar*.
3. In **Allowed Currencies**, select *Euro*.



The screenshot shows the 'Configuration' page for 'RADIAL'. The left sidebar has a menu with 'Currency Setup' selected. The main content area is titled 'Currency Options' and contains three sections: 'Base Currency' (set to 'US Dollar'), 'Default Display Currency' (set to 'US Dollar'), and 'Allowed Currencies' (a list including 'Euro', 'Falkland Islands Pound', 'Fijian Dollar', 'Gambian Dalasi', 'Georgian Kupon Lari', 'Georgian Lari', 'Ghanaian Cedi', and 'Gibraltar Pound'). Each section has a 'Use system value' checkbox.

### Configure Radial Bank Transfer

There are basic configurations that apply to both iDEAL and Sofort bank transfers. To make these settings:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Bank Transfer Configuration**.
2. In **Environment**, select *Sandbox* or *Production*.  
If you are using credentials from your sandbox account, select *Sandbox*.
3. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection.

Note that the store code, API keys, and URLs are unique to each store. Your values will be different from the ones shown in the examples in the section. Please contact your Radial team for the specific configuration values for your store.

- **API URL** example: *https://uat01-epapi-na.gsipartners.com*
- **API Key** example: *9j9g2aSODbG6MAdSHKxgx57wMbyGh11Y*

- **Bank Transfer Payment Action:** Enter the action value as specified by Radial.  
Example: *payments/banktransfer*
  - **Bank Transfer xmlns attribute:** Enter this value as specified by Radial.  
Example: *http://api.gsicommerce.com/schema/checkout/1.0*
  - **Bank Transfer payment retry attempts#:** The number of times to retry a failed bank transfer payment.  
Example: 2
  - **Bank Transfer confirm funds retry attempts#:** The number of times to retry a failed bank transfer confirm funds call.  
Example: 2
  - **Save Log, Days:** The number of days to retain log files.  
Example: 30
4. Copy the value in **Confirm Funds Webhook URL** and provide it to your Radial contact. Bank Transfer uses asynchronous messaging for confirm funds calls, and Radial must configure your store's Webhook URL so that these calls can be enabled at Radial's end.

### Radial Bank Transfer Configuration

Environment <small>[website]</small>	<input type="text" value="Sandbox"/>
Dev API URL <small>[website]</small>	<input type="text" value="https://uat01-epapi-na.gsipartners.com/v1.0/stores/"/> <small>Dev Api Url (https://test.gsipartners.com/v1.0/stores/)</small>
Dev API Key <small>[website]</small>	<input type="text" value="9j9g2aSODbG6MAdsHKxgX57wMbyGh11Y"/> <small>Bank Transfer Dev Api Key</small>
Bank Transfer Payment Action <small>[website]</small>	<input type="text" value="payments/banktransfer"/> <small>Bank Transfer Payment Action Name(payments/banktransfer)</small>
Bank Transfer xmlns attribute <small>[website]</small>	<input type="text" value="http://api.gsicommerce.com/schema/checkout/1.0"/> <small>Bank Transfer xmlns attribute(http://api.gsicommerce.com/schema/checkout/1.0)</small>
Bank Transfer payment retry attempts# <small>[website]</small>	<input type="text" value="3"/> <small>Provided # of times for retry</small>
Bank Transfer confirm funds retry attempts# <small>[website]</small>	<input type="text" value="6"/> <small>Provided # of times for retry</small>
Save Log, Days <small>[website]</small>	<input type="text" value="60"/> <small>Provided number of days logs will store</small>
Confirm Funds Webhook URL <small>[store view]</small>	<input type="text" value="https://apay01-mag02.gsipartners.com/radial/pub/radialbanktransfer/webhook/asynconfirmfunds/"/>

## Enable iDEAL Bank Transfer Payments

To enable iDEAL bank transfers as a payments option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Bank Transfer iDEAL**.
2. In **Enable iDEAL**, select **Yes**.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- In **Title**, enter a text label for the Radial iDEAL Payment Flow.  
Recommended Value: *Radial Bank Transfer iDEAL*
- In **Payment from Applicable Countries**, select *Specific Countries*, since the payment method is supported only for specific countries.
- In **Payment from Specific Countries**, select the countries where the iDEAL payment method will be offered in the checkout page. Example: *Netherlands*

## Enable Sofort Bank Transfer Payments

To enable Sofort bank transfers as a payments option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Bank Transfer Sofort**.
2. In **Enable Sofort**, select **Yes**.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

3. In **Title**, enter a text label for the Radial Sofort Payment Flow.  
Recommended value: *Radial Bank Transfer Sofort*
4. In **Payment from Applicable Countries**, select *Specific Countries*, since the payment method is supported only for specific countries.
5. In **Payment from Specific Countries**, Select the countries where the iDEAL payment method will be offered in the checkout page. Example: *Netherlands*
6. Save the configuration.
7. Perform a cache flush.

⌂ Radial Bank Transfer iDEAL

Enable iDEAL [website]

Bank Transfer is only applicable for netherlands country.

Title For iDEAL [store view]

Payment from Applicable Countries [website]

Use system value

Payment from Specific Countries [website]

- Afghanistan
- Åland Islands
- Albania
- Algeria
- American Samoa
- Andorra
- Angola
- Anguilla
- Antarctica
- Antigua & Barbuda

⌂ Radial Bank Transfer Sofort

Enable SOFORT [website]

Bank Transfer is only applicable for netherlands country.

Title For SOFORT [store view]

Payment from Applicable Countries [website]

Use system value

Payment from Specific Countries [website]

- Afghanistan
- Åland Islands
- Albania
- Algeria
- American Samoa
- Andorra
- Angola
- Anguilla

# Configure Alipay Payment

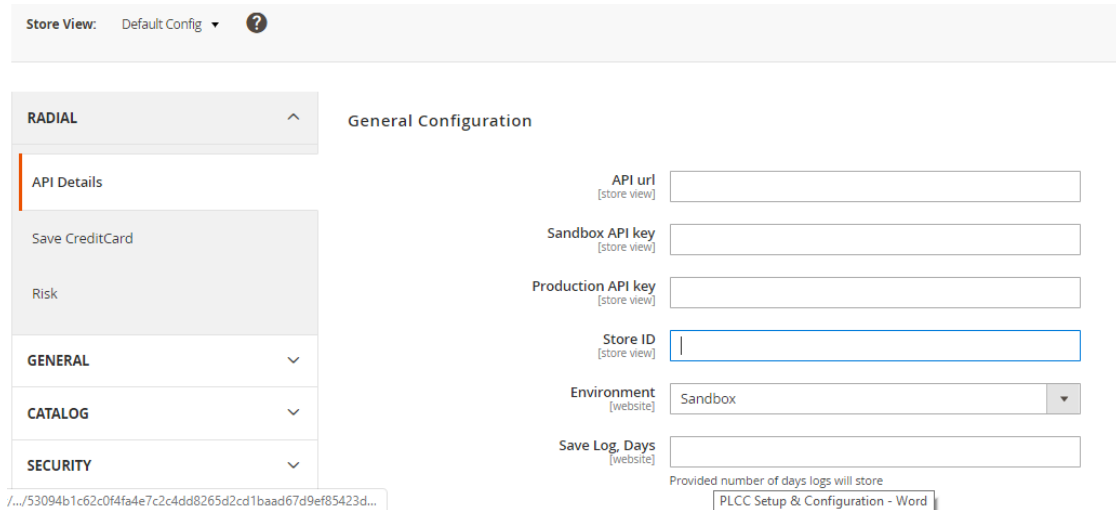
## Complete General Configurations for Alipay

There are some general configurations that must be set before you use Alipay. To complete these configurations:

1. Navigate to **Store > Configuration > Radial > API Details**.
2. In **Environment**, select *Sandbox* or *Production*  
If you are using credentials from your sandbox account, select *Sandbox*.
3. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection.

Note that the store code, API keys, and URLs are unique to each store. Your values will be different from the ones shown in the examples in the section. Please contact your Radial team for the specific configuration values for your store.

- **API URL** example: *https://uat01-epapi-na.gsipartners.com*
- **API Key** example: *9j9g2aSODbG6MAAdsHKxgx57wMbyGh11Y*
- In **Store ID**, enter the unique Store ID value provided by Radial for your store.
- In **Save Log, Days**, enter the number of days to retain log files. For example, *30*.



The screenshot shows a web interface for configuring Alipay. At the top, there's a 'Store View' dropdown set to 'Default Config'. Below this is a sidebar with a 'RADIAL' menu containing 'API Details', 'Save CreditCard', and 'Risk'. The main area is titled 'General Configuration' and contains several input fields: 'API url' (with a 'store view' link), 'Sandbox API key' (with a 'store view' link), 'Production API key' (with a 'store view' link), 'Store ID' (with a 'store view' link), 'Environment' (a dropdown menu currently set to 'Sandbox'), and 'Save Log, Days' (with a 'website' link). A note below the 'Save Log, Days' field states 'Provided number of days logs will store'. At the bottom of the page, there's a breadcrumb trail: 'PLCC Setup & Configuration - Word'.

## Enable Alipay Payments

To enable Alipay as a payment option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Alipay**.
2. In **Enable Alipay**, select **Yes**.

Values for the following fields will populate automatically when the module is installed. If necessary, enter

the following default values or change the values to suit your store's requirements.

- **Title:** a text label for the Radial Alipay Payment Flow.  
Recommended Value: *Radial Alipay*
- **Configured Number of Re-attempts during Timeout Retry during Alipay GetPayment call:** The number of retries to make if Alipay GetPayment encounters a timeout.  
Example: 2
- **Configured Number of Re-attempts during Timeout Retry during Alipay PaymentStatus call:** The number of retries to make if Alipay PaymentStatus encounters a timeout.  
Example: 2
- **Save Log Days:** The number of days to retain log files. Example: 30
- **Payment xmlns attribute:** Enter this value as specified by Radial.  
Example: *http://api.gsicommerce.com/schema/checkout/1.0*

3. Save the configuration.

4. Perform a cache flush.

## Configuration

Shipping Settings	Radial Credit Card
Multishipping Settings	Radial Alipay
Shipping Methods	Enabled <input type="text" value="Yes"/>
Google API	Title <input type="text" value="Radial Alipay"/>
Payment Methods	Configured Number of Re-attempts during Timeout Retry during Alipay Get Payment call <input type="text" value="1"/>
Fraud Protection	Configured Number of Re-attempts during Timeout Retry during Alipay Payment Status call <input type="text" value="1"/>
Gift Cards	Save Log Days <input type="text" value="2"/> Provided number of days logs will store
ENGAGEMENT CLOUD	Payment xmlns attribute <input type="text" value="http://api.gsicommerce.com/schema/checkout/1.0"/>

## Configure OpenInvoice Payments

The OpenInvoice payment method can be used only in specific countries where it is supported, such as Germany and The Netherlands. The store should be set to allow Euro as a currency, with US Dollar as base currency.

### Add Euro to Allowed Currencies

There is a global configuration, Allowed Currencies, where you can select the allowed currencies for the store.

1. Navigate to **Store > Configuration > General > Currency Setup**.
2. In **Base Currency**, select *US Dollar*.
3. In **Allowed Currencies**, select *Euro*.


### Complete General Configurations for OpenInvoice


There are some general configurations that must be set before you can use OpenInvoice. To complete these configurations:

1. Navigate to **Store > Configuration > Radial > API Details**.
2. In **Environment**, select *Sandbox* or *Production*  
If you are using credentials from your sandbox account, select *Sandbox*.
3. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection.

Note that the store code, API keys, and URLs are unique to each store. Your values will be different from the ones shown in the examples in the section. Please contact your Radial team for the specific configuration values for your store.

- **API URL** example: *https://uat01-epapi-na.gsipartners.com*
- **API Key** example: *9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y*
- In **Store ID**, enter the unique Store ID value provided by Radial for your store.
- In **Save Log, Days**, enter the number of days to retain log files. For example, *30*.


Store View: Default Config 


**RADIAL**  General Configuration


API Details

Save CreditCard

Risk

**GENERAL** 

**CATALOG** 

**SECURITY** 

API url [store view]

Sandbox API key [store view]

Production API key [store view]

Store ID [store view]

Environment [website]

Save Log, Days [website]

Provided number of days logs will store

PLCC Setup & Configuration - Word

## Enable OpenInvoice Payments

To enable OpenInvoice as a payments option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial OpenInvoice**.
2. In **Enable OpenInvoice**, select **Yes**.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- **Title:** a text label for the Radial OpenInvoice Payment Flow.  
Recommended Value: *Radial OpenInvoice*
- In **Payment from Applicable Countries**, select *Specific Countries*, since the payment method is supported only for specific countries.
- In **Payment from Specific Countries**, Select the countries where the OpenInvoice payment method will be offered in the checkout page. Example: *Germany*
- **Payment xmlns attribute:** Enter this value as specified by Radial.  
Example: *http://api.gsicommerce.com/schema/checkout/1.0*
- **Dunning Pause Pending in Days:** The number of days for OpenInvoice Dunning Pause  
Example: 5
- **Configured Number of Re-attempts during Timeout Retry during OpenInvoiceAuthorization call:** The number of retries to make in case of timeout.  
Example: 2



Radial Open Invoice

Enabled [website] Yes

Title [store view] Radial Open Invoice

Payment from Applicable Countries [website] Specific Countries  Use system value

Payment from Specific Countries [website]

- Afghanistan
- Åland Islands
- Albania
- Algeria
- American Samoa
- Andorra
- Angola
- Anguilla
- Antarctica
- Antigua & Barbuda

Payment xmlns attribute [store view] http://api.gsicommerce.com/schema/checkout/1.0

Dunning Pause Period in Days [store view] 5

- **Save Log, Days:** The number of days to retain log files.

Example: 30

3. Save the configuration.
4. Perform a cache flush.

## Configure Google Pay Payments

To enable Google Pay as a payment option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Google Pay**.
2. In **Enabled**, select **Yes**.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- **Title:** a text label for the Radial Google Pay Payment Flow.  
Recommended Value: *Radial Google Pay*
  - **Configured Number of Re-attempts during Timeout Retry during Google Pay GetPayment call:** The number of retries to make if GetPayment encounters a timeout.  
Example: 2
  - **Configured Number of Re-attempts during Timeout Retry during Google PayPayment Status call:** The number of retries to make if Payment Status encounters a timeout.  
Example: 2
  - **Save Log, Days:** The number of days to retain log files.  
Example: 30
  - **Payment xmlns attribute:** Enter this value as specified by Radial.  
Example: *http://api.gsicommerce.com/schema/checkout/1.0*
  - **Gateway Id:** Identifier for the gateway to use for Google Pay payments.  
Example: *radialpayments*
  - **Environment:** Choose *Test* or *Production*.
  - **Radial Test Merchant Id:** The Merchant Id for testing, as provided by Radial.  
Example: *MAGT2US*
  - **Radial Production Merchant Id:** The Merchant Id for production, as provided by Radial.
3. Save the configuration.
  4. Perform a cache flush.

⏪ **GooglePay**

Enabled [website] Yes  Use system value

Title [store view] Radial Google Pay  Use system value

Gateway Id [store view] radialpayments

Credit Card Types [website]

- American Express
- Visa
- MasterCard
- Discover
- JCB
- Switch/Maestro
- Diners
- Solo
- Maestro International
- Maestro Domestic

Configured Number of Re-attempts during Timeout Retry [website] 3

Payment xmlns attribute [store view] http://api.gsicommerce.com/schema/checkout/1.0

Save Log, Days [website] 1

Provided number of days logs will store

Environment [website] Test

Radial Test Merchant Id [store view] MAGT2US

---

## Configure Klarna Payments

To configure the Klarna payment method:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Klarna**.
2. In **Enabled**, select Yes.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- **Title:** a text label for the Radial Klarna Payment Flow.  
Recommended Value: *Radial Klarna*
- **Radial JavaScript Environment:** Select *Radial JavaScript Test Environment* or *Radial JavaScript Production Environment*. The names and values of subsequent fields will change according to your selection.
- **Save Log, Days:** The number of days to retain log files.
- **Store ID:** Enter your store's unique Radial ID. Example: *MAGT2US*
- **Nonce (Test/Production) URL:** The nonce URL for your selected JavaScript environment.
- **Radial JavaScript (Test/Production) URL:** The URL for your selected JavaScript environment.
- **Radial Javascript Username:** The username provided by Radial for the nonce generation call for the selected environment.
- **Radial Javascript Password:** The password provided by Radial for the nonce generation call for the selected environment.
- **API URL:** The API URL provided by Radial.  
Example: *https://uat01-epapi-na.gsipartners.com/*
- **Configured Number of Re-attempts during Timeout Retry:** The number of retries to attempt if a Klarna PaymentStatus call encounters a timeout.  
Example: *2*
- **Klarna InstantFinance Auth (Test/Production) URL:** The Klarna URL for your selected environment.
- **Radial API Key (Test/Production):** The API key for your selected environment.  
Example: *9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y*

⊖ **Radial Klarna**

Enabled [website] Yes

Title [store view] Radial Klarna

Save Log, Days [website] 60

Provided number of days logs will store

Radial Javascript Environment [website] Radial JS Test Environment

Radial Klarna Store ID Test [store view] MAGT2US

Nonce Test URL [store view] https://tst.payments.radial.com/hosted-payments/auth/nonce

Radial JS Test URL [store view] https://tst.payments.radial.com/hosted-payments/radial\_klarna\_payments.js

Radial JS Username [store view] MAGT2US

Radial JS Password [store view] .....

Configured Number of Re-attempts during Timeout Retry [website] 3

Radial InstantFinance Auth Test URL [store view] https://uat01-epapi-na.gsipartners.com/

Test URL (https://test.gsipartners.com/)

Radial API Key Test key [store view] 9j9g2aSODbG6MAdsHKxg57wMbyGh11Y

## Configure Interac Online Payments

The Interac Online payment method can be used only in specific countries where it is supported, such as Canada. The store should be set to allow Canadian Dollar as a currency, with US Dollar as base currency.

### Add Canadian Dollar to Allowed Currencies

There is a global configuration, Allowed Currencies, where you can select the allowed currencies for the store.

1. Navigate to **Store > Configuration > General > Currency Setup**.
2. In **Base Currency**, select *US Dollar*.
3. In **Allowed Currencies**, select *Canadian Dollar*.

### Enable Interac Online Payments

To enable OpenInvoice as a payments option:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Interac Online Payment**.
2. In **Enabled**, select Yes.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- **Title:** a text label for the Radial Interac Online Payment Flow.  
Recommended Value: *Radial Interac Online Payment*
- In **Environment**, select *Sandbox* or *Production*. The names and values of subsequent fields will change according to your selection.
- **Dev/Production API URL:** Enter the Dev (Sandbox) or Production API URL provided by Radial.
- **Dev/Production API Key:** Enter the Dev (Sandbox) or Production API key provided by Radial.
- Copy the value in **Funded URL** and provide it to your Radial contact. Radial must configure your store's Funded URL to support Interac Online processing at Radial.
- Copy the value in **Non-Funded URL** and provide it to your Radial contact. Radial must configure your store's Non-Funded URL to support Interac Online processing at Radial.
- In **Interac Online Payment Action**, enter this recommended value: *payments/interaonline/*  
This string will be used as part of a URL like the following example:

```
/v[M.m]/stores/{StoreID}/payments/interaonline/request/{TenderType}.{format}
```

- In **Interac Online Xmlns Attribute**, enter this recommended value:  
*http://api.gsicommerce.com/schema/checkout/1.0*
- In **Payment from Applicable Countries**, select *Specific Countries*, since the payment method is supported only for specific countries.

- In **Payment from Specific Countries**, Select the countries where the Interac Online payment method will be offered in the checkout page.

Example: *Canada*

- **Save Log, Days:** The number of days to retain log files.

Example: 3

- **Interac Online payment retry attempts#:** This many retries will be attempted if an Interac Online payment encounters a timeout.

⊖ Radial Interac Online Payment

Enabled <small>[website]</small>	Yes	▼	
Title <small>[store view]</small>	Radial Interac		
Environment <small>[website]</small>	Sandbox	▼	
Dev API URL <small>[website]</small>	https://uat01-epapi-na.gsispartners.com/v1.0/stores/		
	<small>Dev Api Url (https://test.gsispartners.com/v1.0/stores/)</small>		
Dev API Key <small>[website]</small>	9j9g2aSODbG6MAdsHKxg57wMbyGh11Y		
	<small>Interac Online Dev Api Key</small>		
Funded URL <small>[website]</small>	https://apay01-mag02.gsispartners.com/radial/pub/radialinterac/interac/funded		
Non-Funded URL <small>[website]</small>	https://apay01-mag02.gsispartners.com/radial/pub/radialinterac/interac/nonfunded		
Interac Online Payment Action <small>[website]</small>	payments/interaonline/		
	<small>Interac Online Payment Action Name(payments/interaonline/)</small>		
Interac Online xmlns attribute <small>[website]</small>	http://api.gsicommerce.com/schema/checkout/1.0		
	<small>Interac Online xmlns attribute(http://api.gsicommerce.com/schema/checkout/1.0)</small>		
Payment from Applicable Countries <small>[website]</small>	Specific Countries	▼	<input checked="" type="checkbox"/> Use system value
Payment from Specific Countries <small>[website]</small>	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Afghanistan</p> <p>Åland Islands</p> <p>Albania</p> <p>Algeria</p> <p>American Samoa</p> <p>Andorra</p> <p>Angola</p> <p>Anguilla</p> <p>Antarctica</p> <p>Antigua &amp; Barbuda</p> </div>		
Save Log, Days <small>[website]</small>	60		
	<small>Provided number of days logs will store</small>		
Interac Online payment retry attempts# <small>[website]</small>	5		
	<small>Provided # of times for retry</small>		

# Configure Apple Pay Payments

To enable Apple Pay as a payment method:

1. Navigate to **Store > Configuration > Sales > Payment Methods > Radial Apple Pay**.
2. In Radial Apple Pay Enabled, select **Yes**.

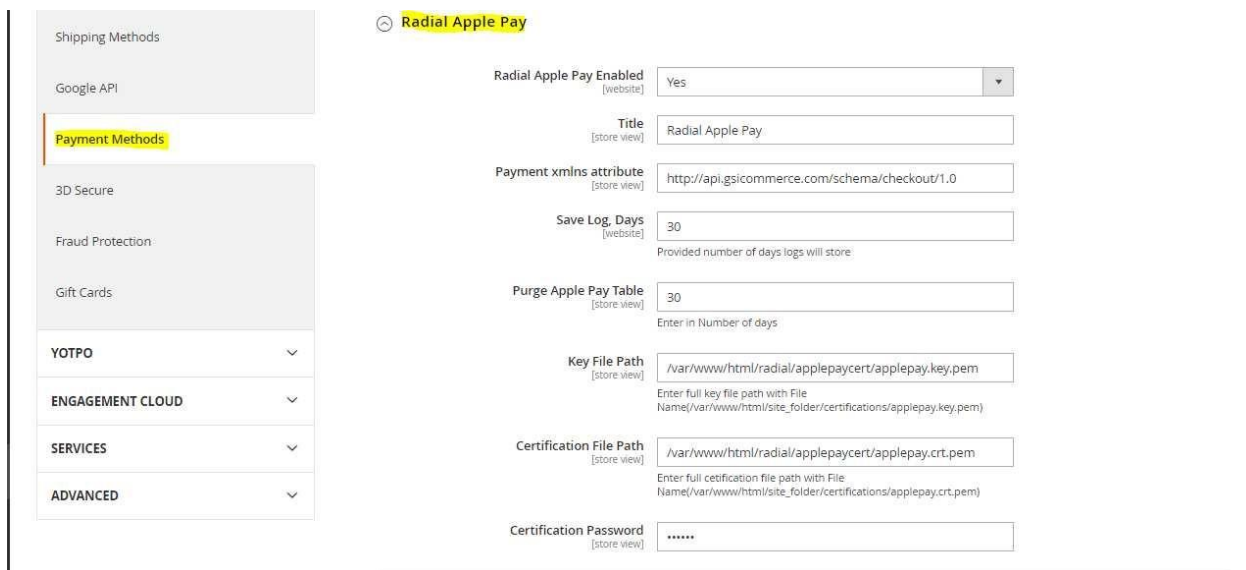
Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

- **Title:** Provide the Title of the Radial Apple Pay Payment which is displayed in checkout page. Recommended: *Radial Apple Pay Payment*.
- **Payment Xmlns:** Recommended value: *http://api.gsicommerce.com/schema/checkout/1.0*
- **Save Log, Days:** The number of days to retain log files. Example: *30*
- **Purge Apple Pay Table:** The number of days after which the apple pay table is purged. Example: *30*
- **Key File Path:** The full file path where your key.pem file is placed on the server.

During your setup with Apple, file.key.pem, along with file.crt.pem, was created from file.p12. The command to generate the key.pem file is:

```
openssl pkcs12 -in Connector_Project_Merchant_Certificate.p12 -out file.key.pem -nocerts -nodes
```

- **Certification File Path:** The full file path where your certification (crt.pem) file is placed on the server.
- **Certification Password:** The password of the certification (crt.pem) file.



**Radial Apple Pay**

- Radial Apple Pay Enabled [website]: Yes
- Title [store view]: Radial Apple Pay
- Payment xmlns attribute [store view]: http://api.gsicommerce.com/schema/checkout/1.0
- Save Log, Days [website]: 30  
Provided number of days logs will store
- Purge Apple Pay Table [store view]: 30  
Enter in Number of days
- Key File Path [store view]: /var/www/html/radial/applepaycert/applepay.key.pem  
Enter full key file path with File Name (/var/www/html/site\_folder/certifications/applepay.key.pem)
- Certification File Path [store view]: /var/www/html/radial/applepaycert/applepay.crt.pem  
Enter full certification file path with File Name (/var/www/html/site\_folder/certifications/applepay.crt.pem)
- Certification Password [store view]: .....



# Configure Link Money Payments

**Restriction:** This payment method can be used customer has the linkmoney account details(Username/password).

1. Log on as a Magento Admin and navigate to **Stores > Settings > Configuration**.
2. In the left panel, expand **Sales** section and choose **Payment Methods**.Select **Sales > Payment Methods > Radial Link money**.
3. In **Title**, enter a text label to identify this payment method during checkout.
4. In Environment, select *Sandbox* or *Production*.

If you are using credentials from your sandbox account, select *Sandbox*.

5. Fill in the following details. These field names will vary (and the appropriate values will differ) based on your environment selection:
  - API URL example: *https://example.com/v1.0/stotes/*

**Save Config**

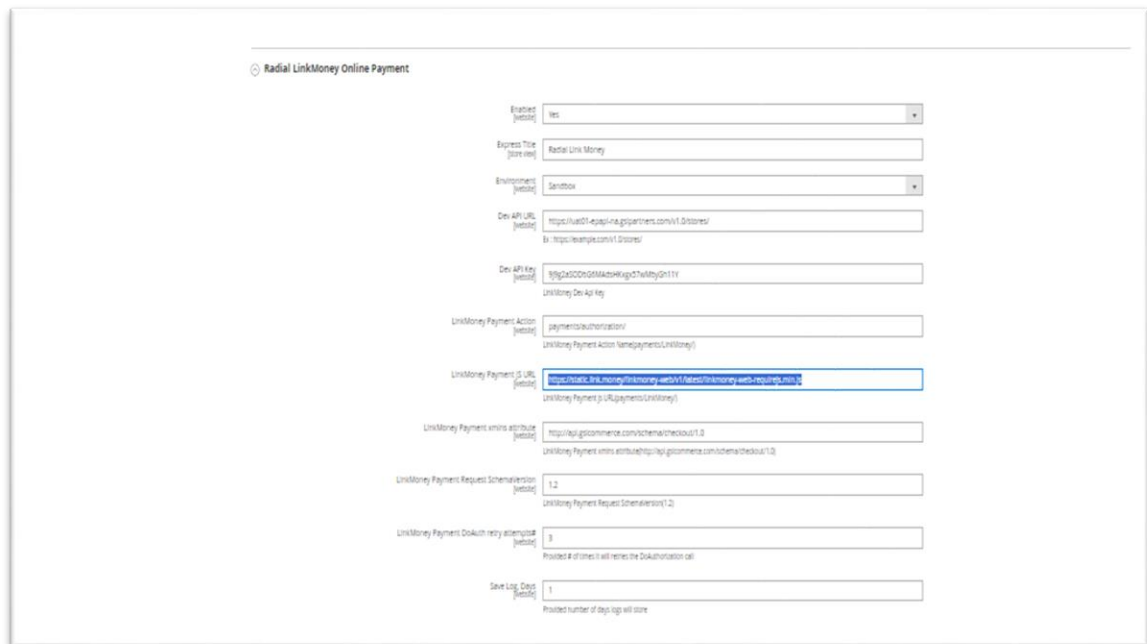
Radial LinkMoney Online Payment

Enabled <small>(website)</small>	Yes
Express Title <small>(store view)</small>	Radial Link Money
Environment <small>(website)</small>	Sandbox
Dev API URL <small>(website)</small>	https://uat01-epapi-na.gsipartners.com/v1.0/stotes/ <small>Ex : https://example.com/v1.0/stotes/</small>
Dev API Key <small>(website)</small>	9j9g2aSODbG6MAdsHKxgX57wMbyGh11Y <small>LinkMoney Dev Api Key</small>
LinkMoney Payment Action <small>(website)</small>	payments/authorization/ <small>LinkMoney Payment Action Name(payments/LinkMoney)</small>
LinkMoney Payment JS URL <small>(website)</small>	https://static.link.money/linkmoney-web/v1/latest/linkmoney-web-requirejs.min.js <small>LinkMoney Payment JS URL(payments/LinkMoney)</small>
LinkMoney Payment xmlns attribute <small>(website)</small>	http://api.gsicommerce.com/schema/checkout/1.0 <small>LinkMoney Payment xmlns attribute(http://api.gsicommerce.com/schema/checkout/1.0)</small>
LinkMoney Payment Request SchemaVersion <small>(website)</small>	1.2 <small>LinkMoney Payment Request SchemaVersion(1.2)</small>
LinkMoney Payment DoAuth retry attempts# <small>(website)</small>	3 <small>Provided # of times it will retries the DoAuthorization call</small>
Save Log_Days <small>(website)</small>	1 <small>Provided number of days logs will store</small>

- PI Key example: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

- **Link Money Payment Action:** Value should be *payments/authorization/*
- **Link Money Payment xmlns attribute:** This is a Radial API request attribute. Value should be:  
*http://api.gsicommerce.com/schema/checkout/1.0*
- **Link Money Payment retry attempts#:** When the API returns a Timeout response, this many retries will be attempted internally. Radial suggests a value of 3
- **Save Log, Days:** The number of days to retain log files.
- **Link Money Payment Js URL:** Link money JS url path.  
**Ex :** <https://static.link.money/linkmoney-web/v1/latest/linkmoney-web-requirejs.min.js>

The following screenshot shows default configuration options.



Radial LinkMoney Online Payment

Enabled (boolean)	Yes
Express Title (string)	Radial Link Money
Environment (enum)	Sandbox
Dev API URL (url)	https://uat01-epagi-na.gpipartners.com/v1/Stores/ Ex: https://example.com/v1/Stores/
Dev API Key (string)	9992a00b0868a2b40407467b91111 LinkMoney Dev Api Key
LinkMoney Payment Action (string)	/payments/authorization/ LinkMoney Payment Action Name(payments.LinkMoney/)
LinkMoney Payment Js URL (url)	https://static.link.money/linkmoney-web/v1/latest/linkmoney-web-requirejs.min.js LinkMoney Payment Js URL(payments.LinkMoney/)
LinkMoney Payment xmlns attribute (url)	http://api.gsicommerce.com/schema/checkout/1.0 LinkMoney Payment xmlns attribute(http://api.gsicommerce.com/schema/checkout/1.0)
LinkMoney Payment Request SchemalVersion (string)	1.2 LinkMoney Payment Request SchemalVersion(1.2)
LinkMoney Payment DoAuth retry attempts (integer)	3 Provided # of times it will retry the DoAuthorization call
Save Log, Days (integer)	1 Provided number of days logs will store

Link Money frond-end Workflow:

apay01-maguz.gsipartners.com/radial/pub/checkout/#payment

Radial Apple Pay  
 Radial Google Pay  
 Radial Klarna  
 Radial LinkMoney


My billing and shipping address are the same

Arun Kumar  
1075 1st Avenue666 King of Prussia, Pennsylvania 19406  
United States  
4843407988


**Place Order**

Radial PayPal Express  
 Radial Credit Card


2 Items in Cart


Ship To: 

Arun Kumar  
1075 1st Avenue666  
King of Prussia, Pennsylvania 19406  
United States  
4843407988

Shipping Method: 


Free Shipping - Free


×





**Secure Bank Payment**

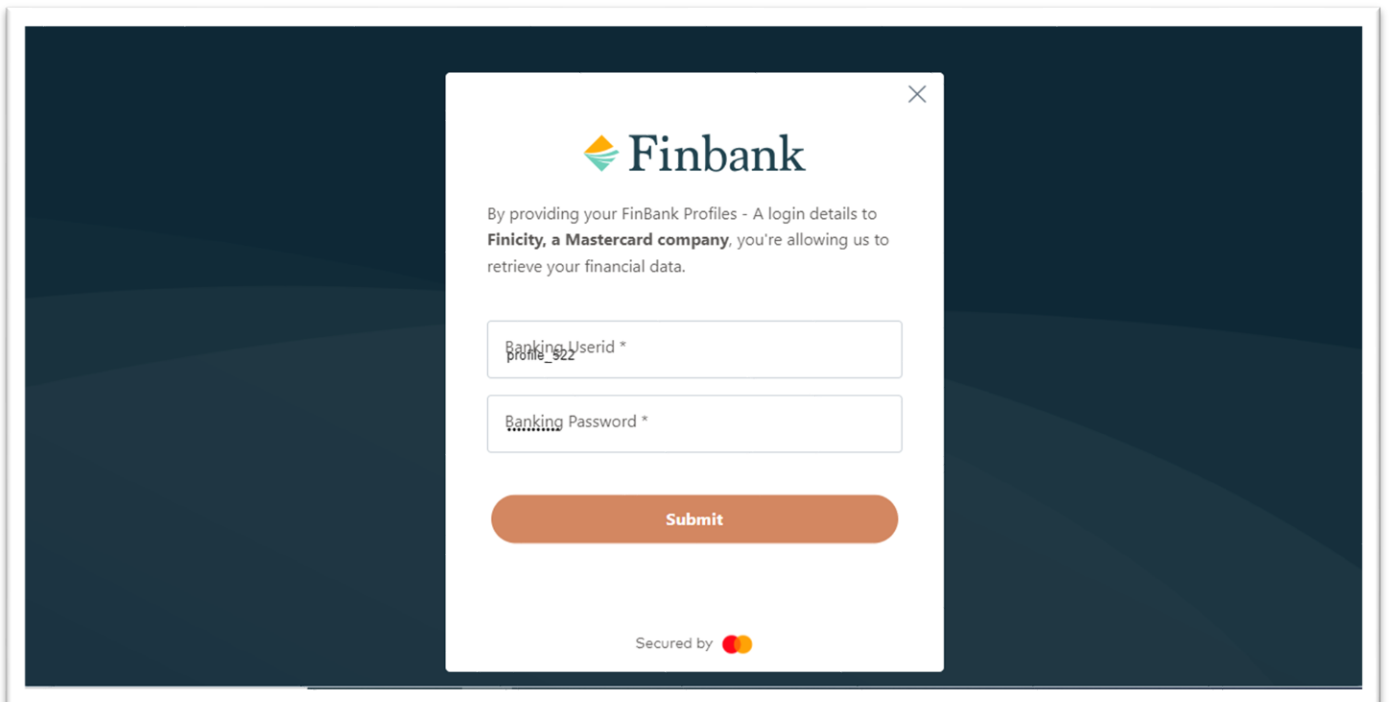
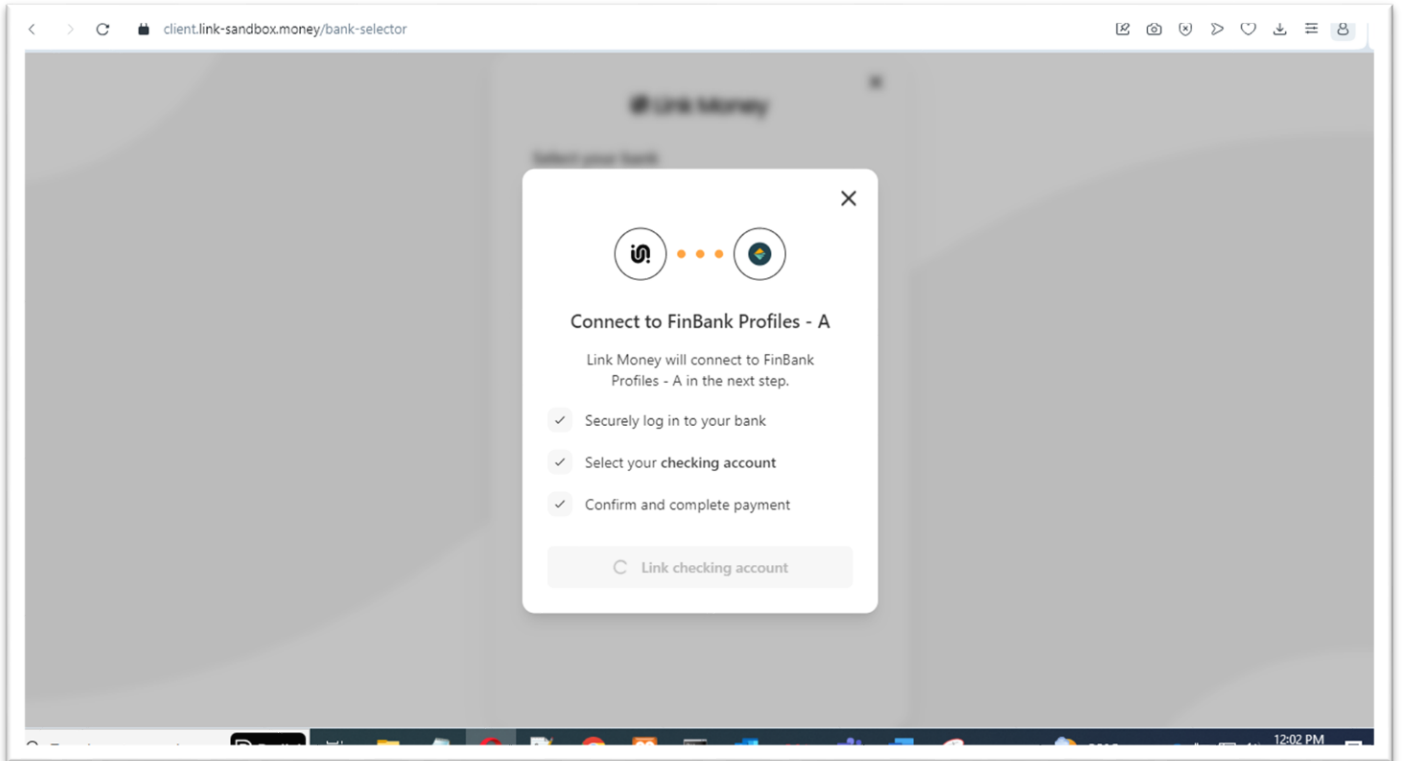
Radial has partnered with Link Money to securely process payments via your checking account.

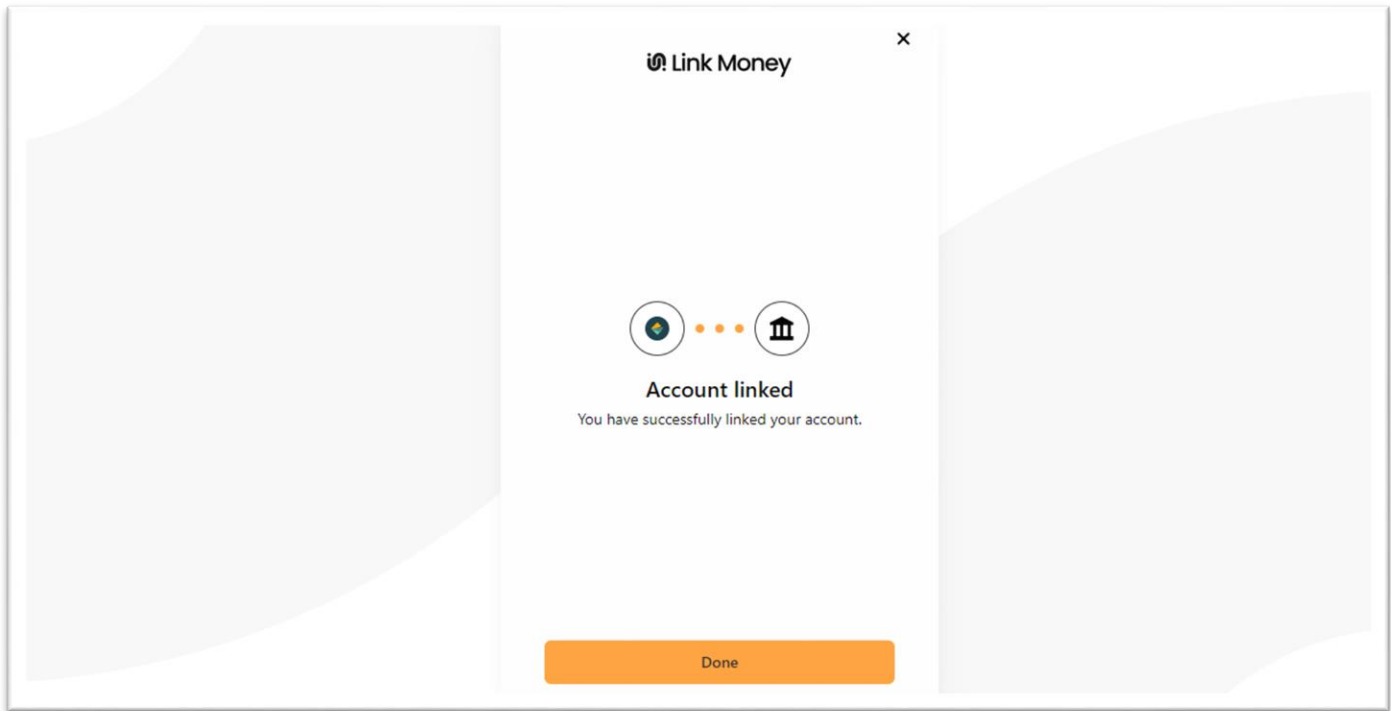
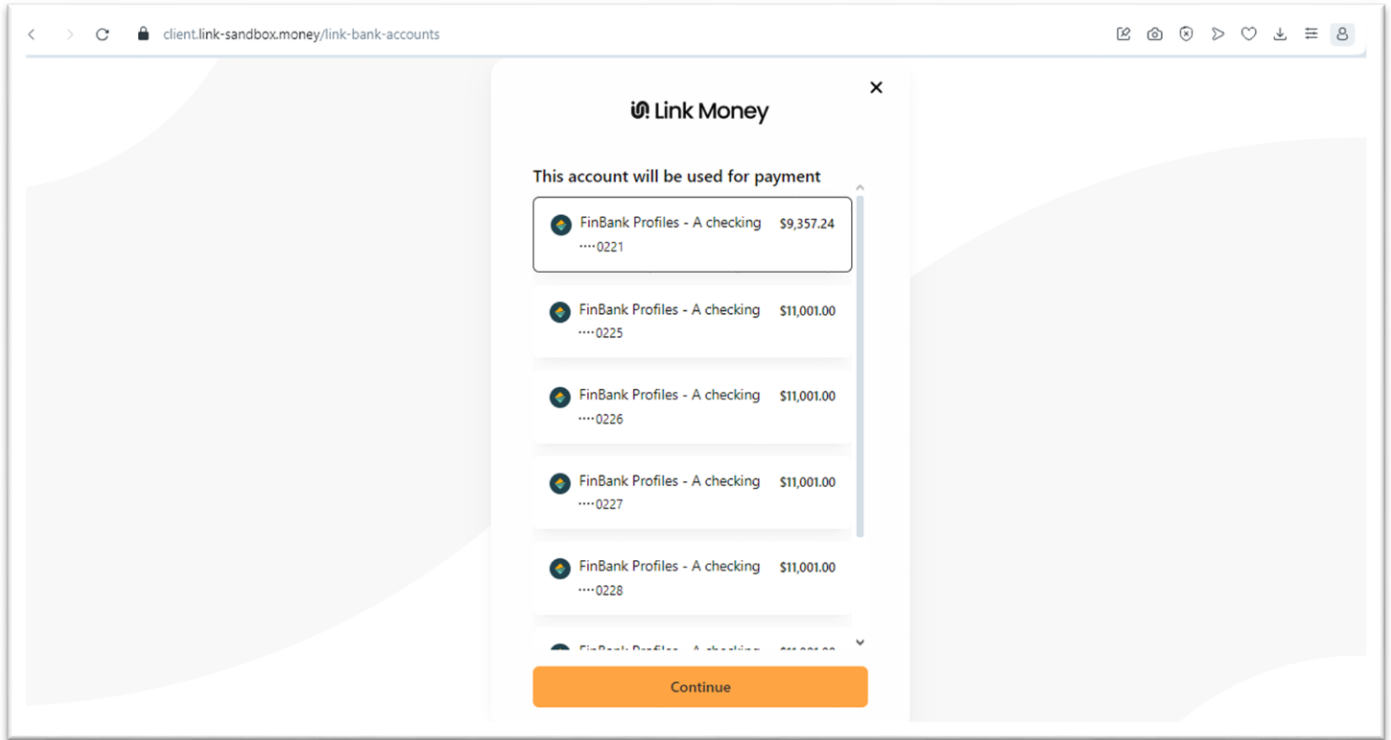
Account and transaction data handling 

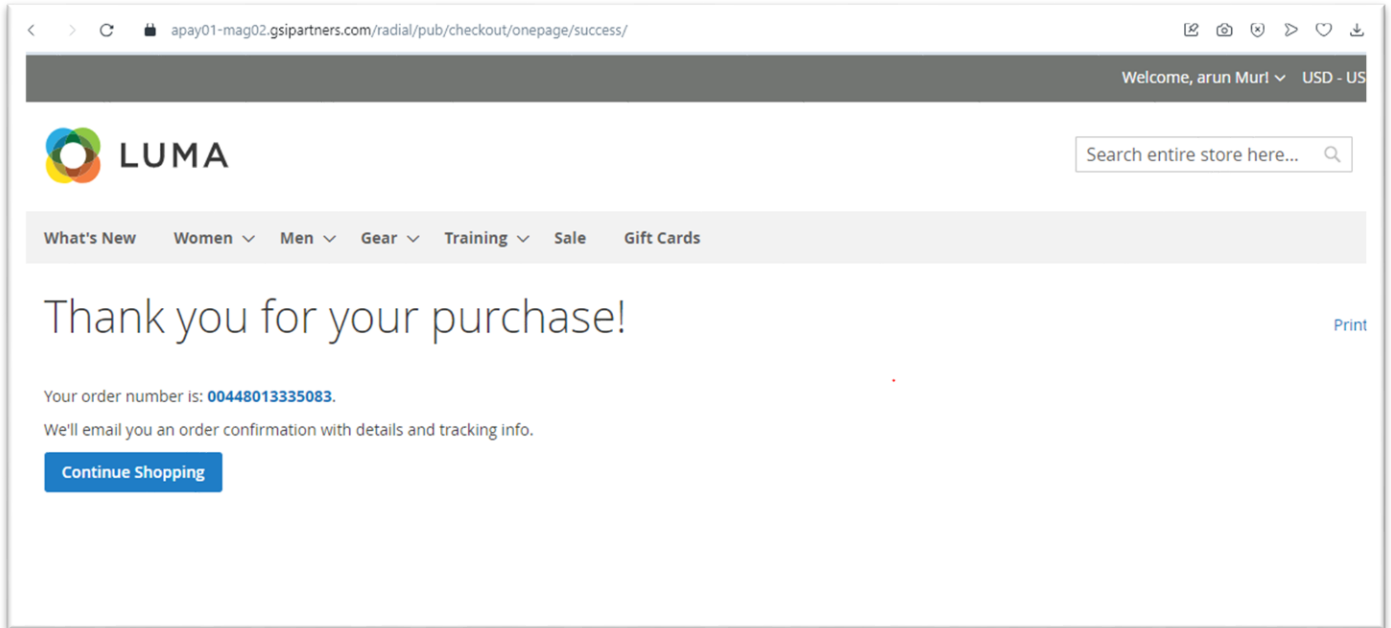
**Get started**

By pressing "Get started", you agree to Pay by Bank's [Terms & Conditions](#), Link Money's [Privacy Policy](#), and Finicity, a Mastercard company, [Terms & Conditions](#) and [Privacy Policy](#).

Access provided by  & 








apay01-mag02.gsispartners.com/radial/pub/checkout/onepage/success/

Welcome, arun Muri | USD - US

 **LUMA**

What's New | Women | Men | Gear | Training | Sale | Gift Cards

# Thank you for your purchase!

[Print](#)

Your order number is: **00448013335083**.

We'll email you an order confirmation with details and tracking info.

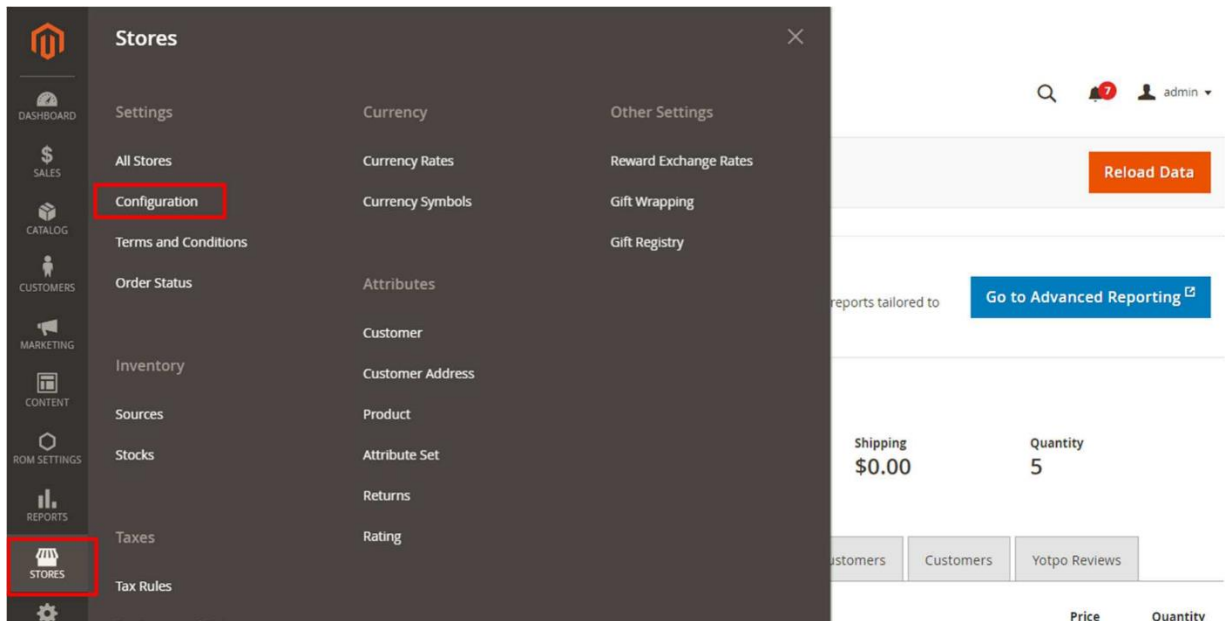
[Continue Shopping](#)

## Section 4: Configure Gift Cards

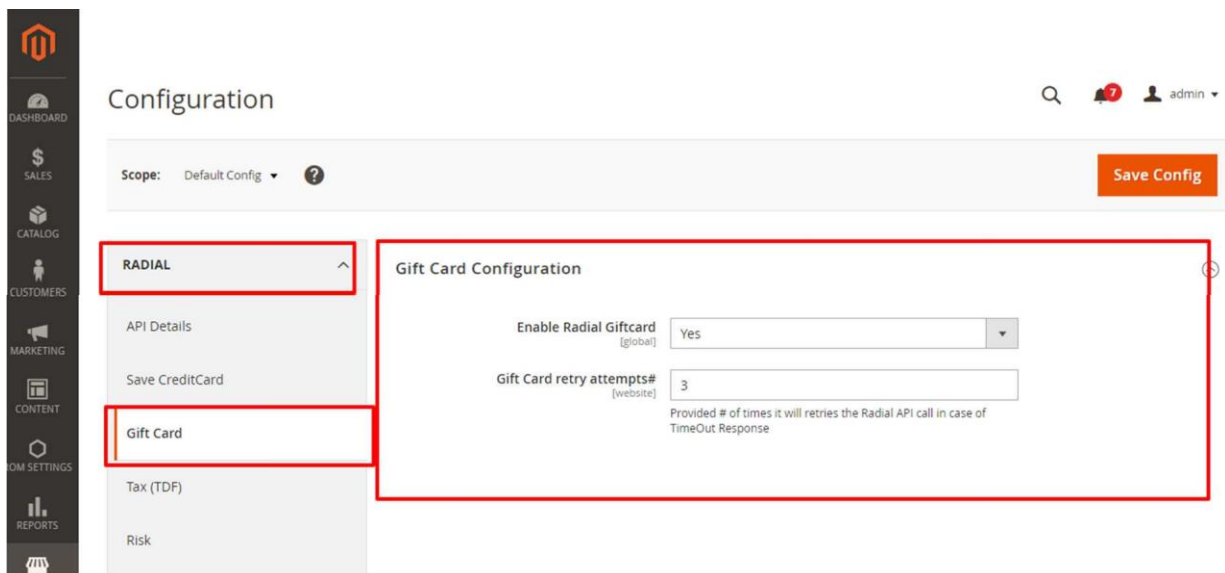
### Enable the Gift Card Module

To enable the Radial Gift Card module:

1. Navigate to **Store > Configuration**.



2. In the Radial section, select the Gift Card tab.



3. In **Enable Radial Giftcard**, select Yes. This enables the Gift Card module.
4. In **Gift Card Retry attempts#**, enter the number of retries to attempt in case a gift card API call encounters a timeout response. Example: 3.

## Configure Magento Gift Card Defaults

To configure default values for Magento gift cards:

1. Navigate to **Store > Configuration**.
2. Navigate to **Sales > Gift Cards**.
3. Specify values in Gift Card Email Settings, including:
  - Gift Card Notification Email Sender
  - Gift Card Notification Email Template
4. Specify values in Gift Card General Settings, including:
  - Select whether the Gift card is redeemable or not
  - Lifetime of GC
  - Allow Gift Card Message
  - Gift Card Message Maximum Length
  - When the Gift Card should generate, either after creation of order or after creation of invoice
5. Specify values in Email Sent from Gift Card Account Management, including:
  - Gift card Email Sender
  - Gift Card Template
6. Specify values in Gift Card Account General Settings, including:
  - Code Length, Code Format, Code Prefix & Suffix

**Click Save Config.**



-  DASHBOARD
-  SALES
-  CATALOG
-  CUSTOMERS
-  MARKETING
-  CONTENT
-  ROM SETTINGS
-  REPORTS

- RADIAL** ▾
- RADIAL ROM ▾
- GENERAL ▾
- CATALOG ▾
- SECURITY ▾
- CUSTOMERS ▾
- SALES** ▴
- Sales
- Sales Emails
- PDF Print-outs
- Tax
- Checkout
- Shipping Settings
- Multishipping Settings
- Shipping Methods
- Google API
- Payment Methods
- Fraud Protection
- 3D Secure
- Gift Cards**
- Address Validation
- YOTPO ▾
- ENGAGEMENT CLOUD ▾
- SERVICES ▾
- ADVANCED ▾

## Configuration

🔍   admin ▾

Scope: Default Config ▾ ?

Save Config

### Gift Card Email Settings ⌵

Gift Card Notification Email Sender [store view]  ▾

Gift Card Notification Email Template [store view]  ▾

### Gift Card General Settings ⌵

Redeemable [global]  ▾

Lifetime (days) [global]   
If empty no expiration.

Allow Gift Message [store view]  ▾

Gift Message Maximum Length [store view]

Generate Gift Card Account when Order Item is [global]  ▾

### Email Sent from Gift Card Account Management ⌵

Gift Card Email Sender [store view]  ▾

Gift Card Template [store view]  ▾

### Gift Card Account General Settings ⌵

Code Length [global]   
Excluding prefix, suffix and separators.

Code Format [global]  ▾

Code Prefix [global]

Code Suffix [global]

Dash Every X Characters [global]   
if empty no separation

New Pool Size [global]

Low Code Pool Threshold [global]

[global]  
Code Pool used: 0 (free 1000 of 1000 total).

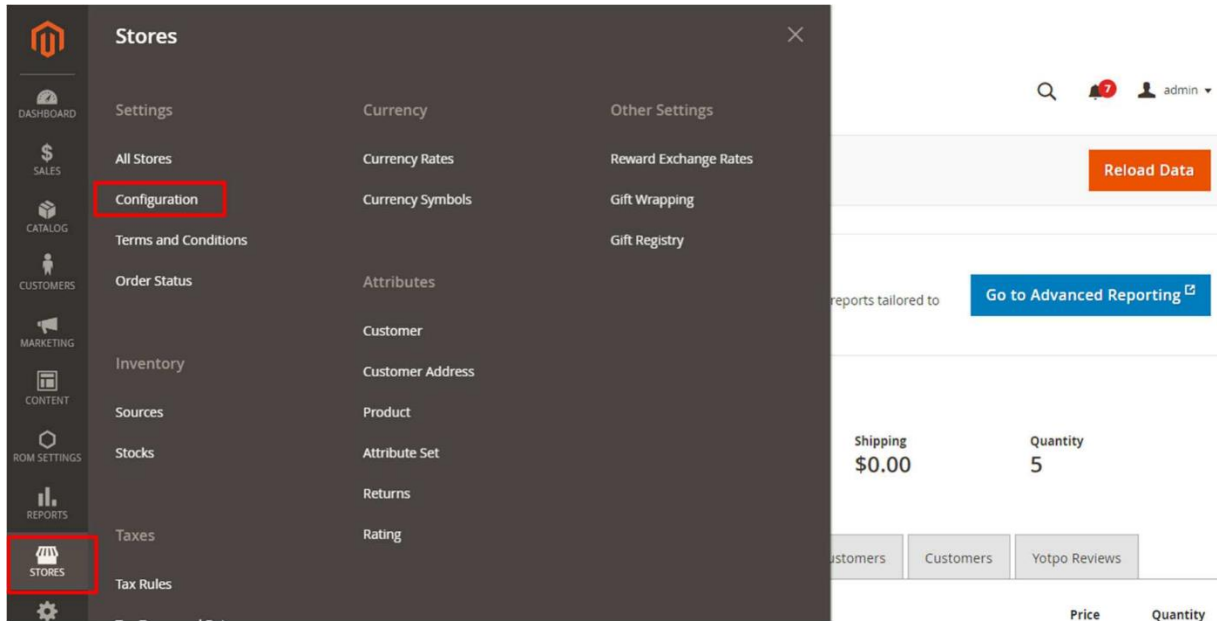
IMPORTANT: Be sure to save your configuration before running generation process.

## Section 5: Configure Tax

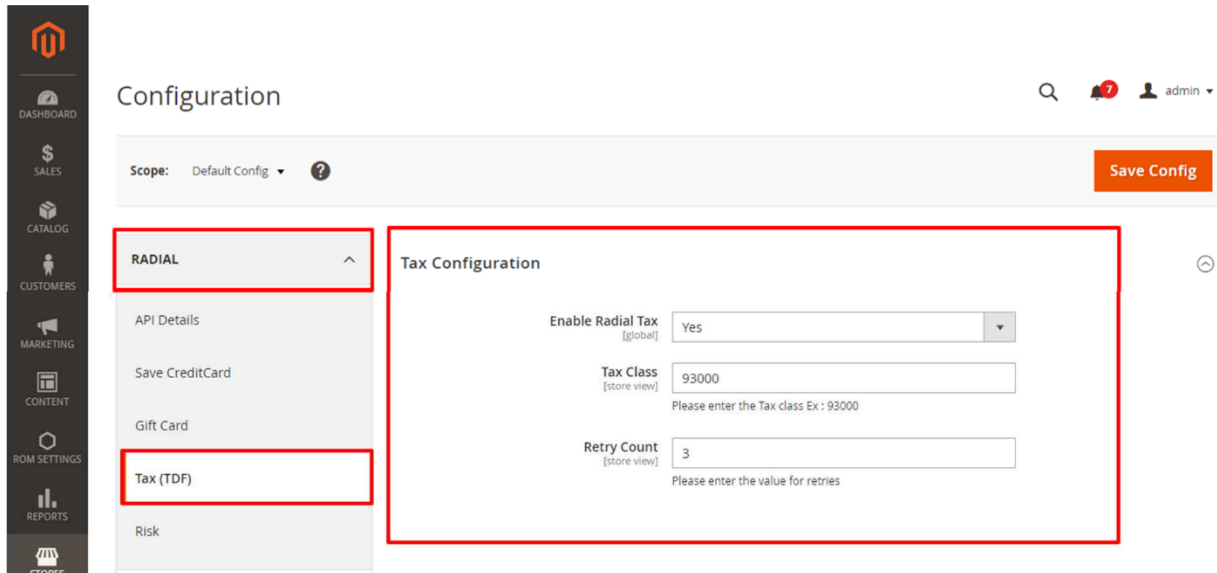
### Enable the Tax Module

To enable the Radial Tax module:

1. Navigate to **Store > Configuration**.



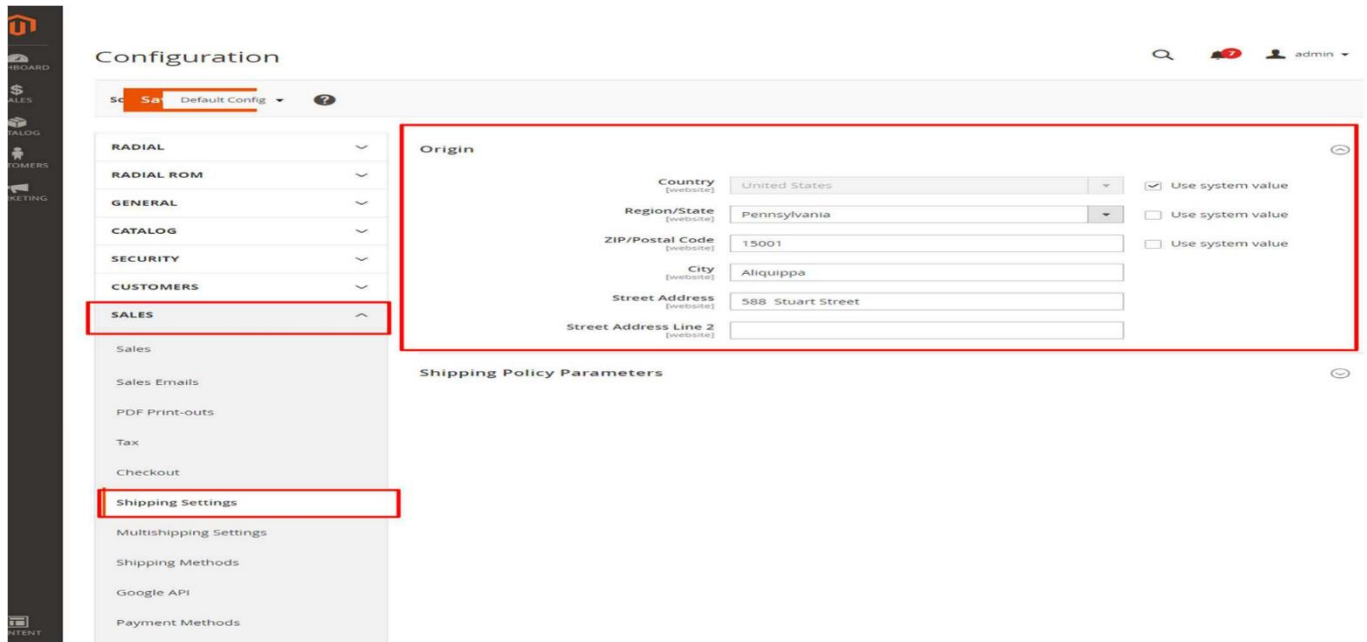
2. In the Radial section, select the Tax (TDF) tab.



3. In **Enable Radial Tax**, select **Yes**.
4. In **Tax Class**, enter the tax class value. This value is for shipping. Example: *93000*.
5. In **Retry Count**, enter the number of retries to attempt for a Tax API call. Example: *3*.

## Configure Origin Settings for Shipping

1. Navigate to **Sales > Shipping Settings**.
2. In the Origin section, enter the address from which products will be shipped. There are fields for address components including Country, Region/State, Zip/Postal Code, City, Street Address, Street Address Line 2.



**Configuration**

SALES Default Config

- RADIAL
- RADIAL ROM
- GENERAL
- CATALOG
- SECURITY
- CUSTOMERS
- SALES**
  - Sales
  - Sales Emails
  - PDF Print-outs
  - Tax
  - Checkout
  - Shipping Settings**
  - Multishipping Settings
  - Shipping Methods
  - Google API
  - Payment Methods

**Origin**

Country [website]   Use system value

Region/State [website]   Use system value

ZIP/Postal Code [website]   Use system value

City [website]

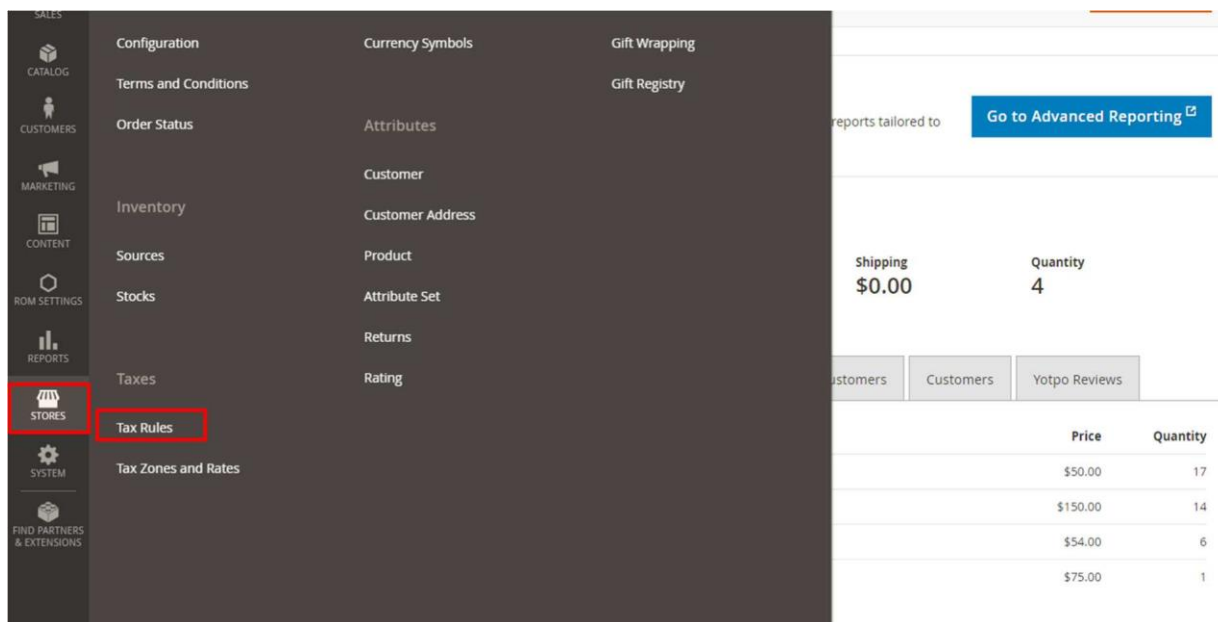
Street Address [website]

Street Address Line 2 [website]

**Shipping Policy Parameters**

## Create Product Tax Classes and Assign to Products

1. Navigate to **Store > Taxes > Tax Rules**.



**SALES**

- Configuration
- Terms and Conditions
- Order Status
- Inventory
- Sources
- Stocks
- Taxes
  - Tax Rules**
  - Tax Zones and Rates
- System
- Find Partners & Extensions

Currency Symbols

Gift Wrapping

Gift Registry

Attributes

Customer

Customer Address

Product

Attribute Set

Returns

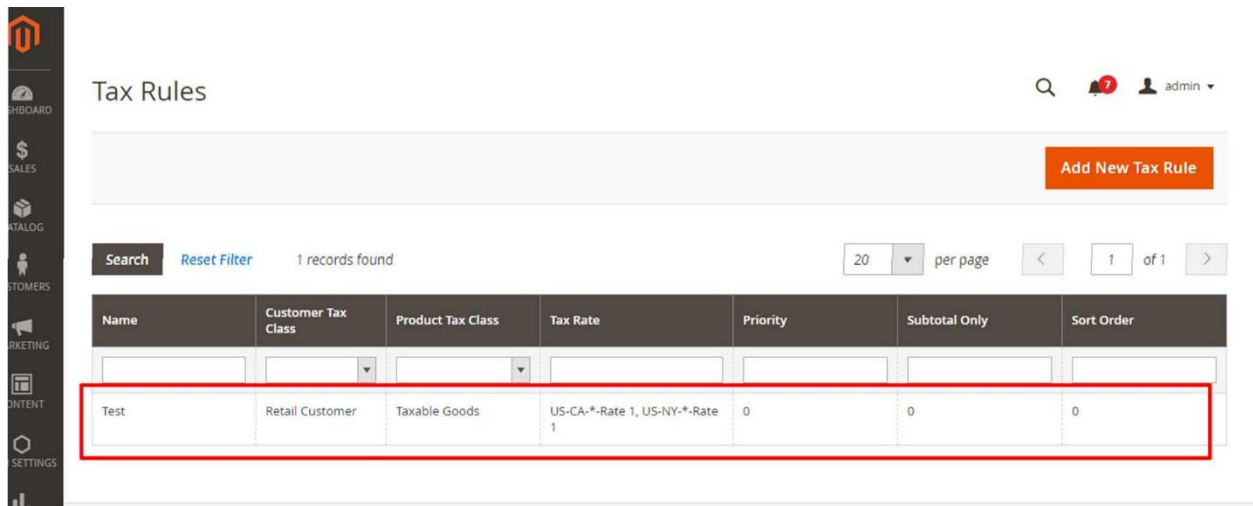
Rating

reports tailored to [Go to Advanced Reporting](#)

Shipping	Quantity
\$0.00	4

Price	Quantity
\$50.00	17
\$150.00	14
\$54.00	6
\$75.00	1

2. Select a tax rule for which you want to create a Product Tax Class.



Tax Rules

Search [Reset Filter](#) 1 records found

20 per page 1 of 1

Name	Customer Tax Class	Product Tax Class	Tax Rate	Priority	Subtotal Only	Sort Order
Test	Retail Customer	Taxable Goods	US-CA-*Rate 1, US-NY-*Rate 1	0	0	0

Add New Tax Rule

3. In the selected rule, click the triangle to expand the Additional Settings section.
4. Under Product Tax Class, click **Add New Tax Class**
5. Select the **Correct** check box.
6. Click **Save Rule**.

[← Back](#) [Delete Rule](#) [Reset](#) [Save and Continue Edit](#) [Save Rule](#)

Tax Rule Information

Name \*

Tax Rate \*

2 selected

US-CA-\*Rate 1

US-NY-\*Rate 1

[Add New Tax Rate](#)

▼ **Additional Settings**

Customer Tax Class \*  Retail Customer

Customer Tax Class \*  Retail Customer

[Add New Tax Class](#)

Product Tax Class \*  Taxable Goods

Refund Adjustments

Gift Options

Order Gift Wrapping

Item Gift Wrapping

Printed Gift Card

Reward Points

76800

[Add New Tax Class](#)

Priority \*

Tax rates at the same priority are added, others are compounded.

Calculate Off Subtotal Only

Sort Order \*

Import Tax Rates [Choose File](#) No file chosen

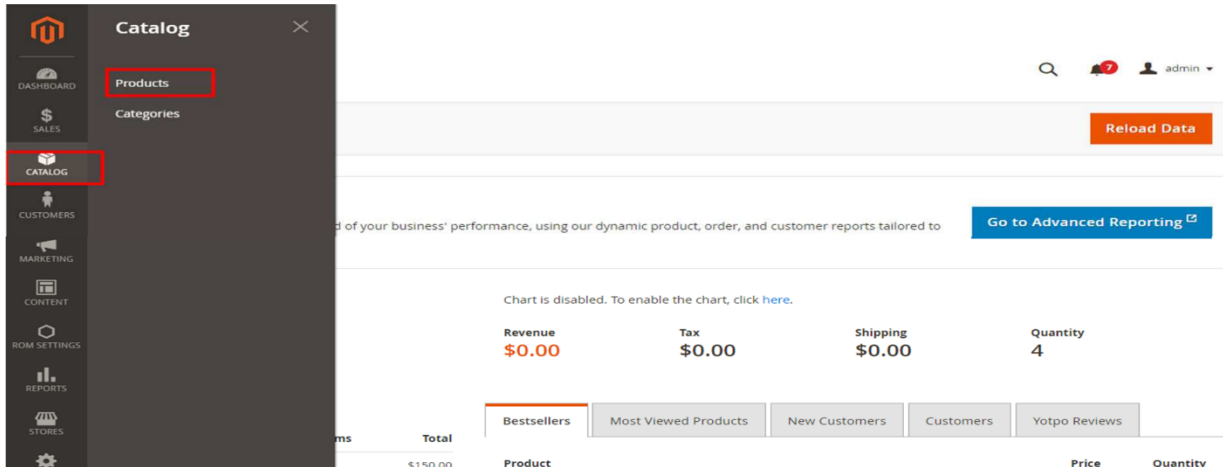
[Import Tax Rates](#)

Export Tax Rates [Export Tax Rates](#)

## Select the Tax Class and County of Manufacture for a Product

To select the product tax class and the country of manufacture for a product:

1. Navigate to **Catalog > Products**.



The screenshot shows the Radial dashboard interface. On the left, a dark sidebar contains navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, ROM Settings, Reports, and Stores. The 'Catalog' menu item is highlighted with a red box. The main content area shows a 'Products' sub-menu also highlighted with a red box. Below the navigation, there are several data cards and a table. The 'Revenue' card shows \$0.00, 'Tax' shows \$0.00, and 'Shipping' shows \$0.00. A 'Quantity' card shows 4. Below these are tabs for 'Bestsellers', 'Most Viewed Products', 'New Customers', 'Customers', and 'Yotpo Reviews'. At the bottom, a table header is visible with columns for 'Product', 'Price', and 'Quantity'.

2. Select a product.
3. In the product details, select a value for **Tax Class**.
4. Select a value for **Country of Manufacture**.

## Push It Messenger Bag

Scheduled Changes Schedule New Update

Scope: All Store Views ? ← Back Add Attribute Save

**Enable Product** [website]  Yes

**Attribute Set** [global] Default

**Product Name** [store view] \* Push It Messenger Bag

**SKU** [global] \* MAGT2NA\_Test1

**Price** [global] \* \$ 50.00  
[Advanced Pricing](#)

**Tax Class** [website] 76800

**Quantity** [global] 1107  
[Advanced Inventory](#)

**Stock Status** [global] In Stock

**Set Product as New**  No

**Weight** [global] 2 lbs This item has weight

**Visibility** [store view] Catalog, Search

**Categories** [global] Radial Products New Category

**Country of Manufacture** [website] India

**Enable RMA** [website]  No  Use Config Settings

**Color** [global]

## Section 6: Configure Radial Fraud

This section details the configurations for the Risk APIs that connect you to Radial's fraud assessment service.

### Configure Risk

To configure Radial's Risk APIs:

1. While logged in as a Magento Admin, navigate to **Stores > Configuration > Radial > Risk**.
2. In **Enable**, select *Yes* to enable the Radial Risk module.
3. In **Environment**, select *Sandbox* or *Production*. If you are using credentials from your sandbox account, select *Sandbox*.

The names and values of subsequent fields will change according to your selection.

4. **(Sandbox/Production) API URL**: The API URL provided by Radial for your selected environment.
5. **(Sandbox/Production) API Key**: The API key provided by Radial for your selected environment.
6. **RDF (Sandbox/Production) API URL**: The RDF API URL provided by Radial for your selected environment. Example: `https://tst-rdf.radial.com/rdf/scripts/v1/rdf.min.js`



- Dashboard
- Sales
- Catalog
- Customers
- Marketing
- Content
- Radial
- ROM Settings
- Reports
- Stores
- System

- RADIAL
- API Details
- Address Validation
- Save CreditCard
- Gift Card
- Tax (TDF)
- Risk**
- RADIAL ROM
- GENERAL
- CATALOG
- SECURITY
- CUSTOMERS
- SALES
- SERVICES
- ADVANCED

### Risk Configuration

**Enable**

**Environment**

**Sandbox API url**   
Sandbox Api Uri (https://uat01-epapi-na.gsipartners.com)

**Sandbox API Key**   
Sandbox Api Key

**RDF Sandbox API URL**   
Sandbox RDF Api Uri (https://tst-rdf.radial.com/rdf/service/scripts?clientIdentifier=)

**Purge Risk Tables**   
Enter in Number of days

**Store ID**   
Store ID

**Purge Risk Error Table**   
Enter in Number of days

**Save Risk Log, Days**   
Provided number of days logs will store

**Email Id**   
Email Id

**Time Interval to get Orders for Risk Status ( in minutes )**   
Time Interval to get Orders for Risk Status ( in minutes )

**Retry Count for Order Status**   
Retry Count for Order Status

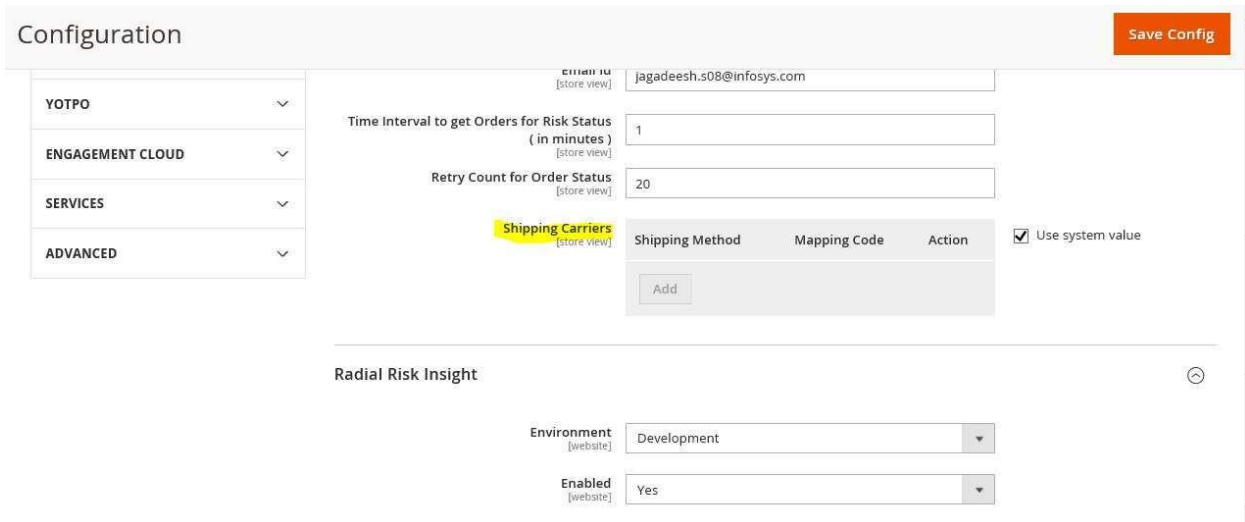
**Shipping Carriers**

Shipping Method	Mapping Code	Action

5. **Purge Risk Tables:** The number of days after which risk log entries will be purged. Example: 30.
6. **ID:** The store's unique ID as provided by Radial.
7. **Purge Risk Error Tables:** The number of days after which risk error log entries will be purged. Example: 30.
8. **Email id:** The risk processing admin email address for your store. If the order status is fraud\_suspend, Magento sends a notification email to this address.
9. **Time interval to get Orders for Risk Status:** A cron job checks the Order Status API at regular intervals for orders with risk processing. Enter a number of minutes for the cron job interval time. Example: 5.
10. **Retry count for Order Status:** The number of retries for the cron job to check a given order with fraud\_suspend status. If the order remains in fraud\_suspend status for this number of retries, the cron job will stop checking for this order ID. Example: 5.
11. In **Shipping Carriers**, map the shipping methods you use to Radial shipping vendor codes.
  - Enter the Magento shipping carrier in the **Shipping Method** column, and enter the corresponding Radial shipping vendor code in the **Mapping Code** column.
  - **Mapping Code** values for carriers should be limited to these five Radial shipping vendor codes: *UPS* (United Parcel Service), *FEDEX* (Federal Express), *USPS* (US Postal Service), *DHL*, and *OTHER* (any other carrier).

- Enter the Magento delivery method (carrier service) in the **Shipping Method** column, and enter a corresponding delivery method code in the **Mapping Code** column.
- **Mapping Code** values for delivery methods are limited to a maximum of 20 characters, and they should be similar to these example values: STANDARD, EXPRESS, 1DAY, 2DAY, 3DAY, SATURDAY, INSTOREPICKUP, SHIPTOSTORE, EMAIL.
- To add another shipping method to the list, click **Add**.
- To delete a shipping method entry, click the trash can icon in the **Action** column.

## 12. Click **Save Config**.



Configuration Save Config

YOTPO

ENGAGEMENT CLOUD

SERVICES

ADVANCED

Time Interval to get Orders for Risk Status (in minutes)

Retry Count for Order Status

**Shipping Carriers**

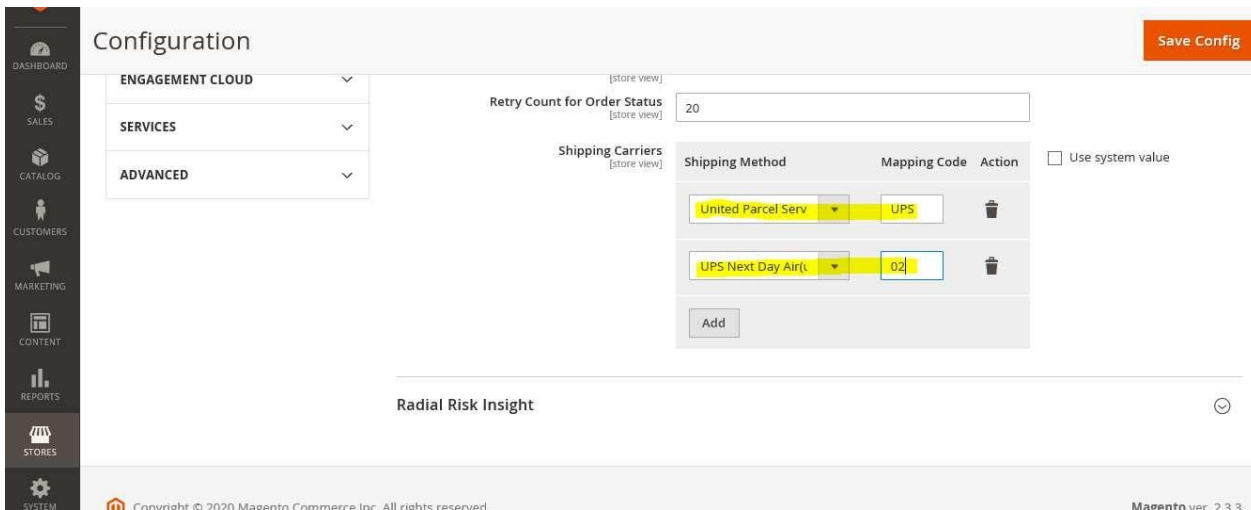
Shipping Method	Mapping Code	Action
Add		

Use system value

Radial Risk Insight

Environment

Enabled



Configuration Save Config

ENGAGEMENT CLOUD

SERVICES

ADVANCED

Retry Count for Order Status

**Shipping Carriers**

Shipping Method	Mapping Code	Action
United Parcel Serv	UPS	
UPS Next Day Air(L)	02	
Add		

Use system value

Radial Risk Insight

Copyright © 2020 Magento Commerce Inc. All rights reserved. Magento ver. 2.3.3

The Risk Order Confirmation message contains a shipping vendor code for the order (Order/LineDetails/ShippingVendorCode), which will include the value that you entered as the Mapping Code for the shipping carrier.

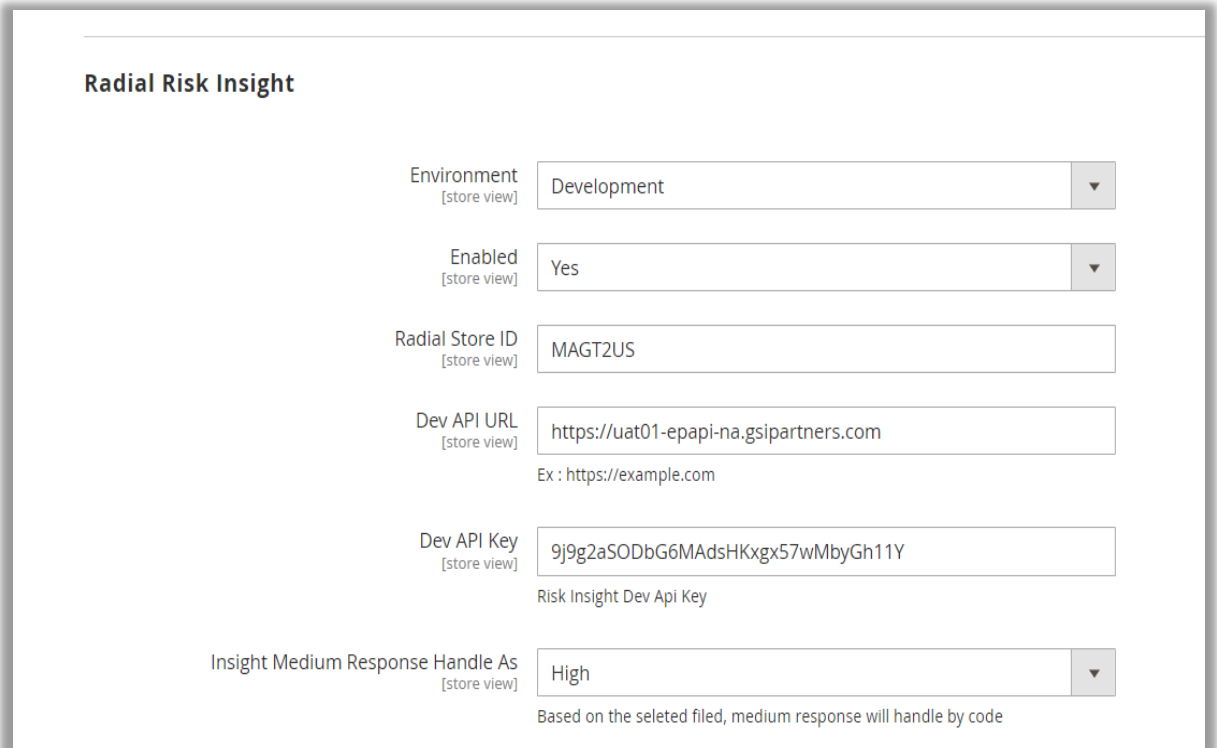
The confirmation message also contains a delivery method (Order/Line Details/DeliveryMethod), which will include the value that you entered as the Mapping Code for Magento delivery method.

## Configure Risk Insight

**Restriction:** This module is available only for payments made with credit card. Risk Insight is also called “Pre-Auth,” and it is a special feature that is independent of the standard Risk module.

To enable and configure Risk Insight:

1. Navigate to **Store > Configuration > Radial > Risk > Radial Risk Insight**.
2. In **Environment**, select *Development* or *Production*. The names and values of subsequent fields will change according to your selection.
3. In **Enabled**, select *Yes* to enable the module and display the Radial Risk Insight option.
4. In **Store ID**, enter the unique Radial ID for your store. Example: *MAGT2US*.
5. **Dev/Production API URL:** The API URL provided by Radial for Development (Sandbox) or Production.
6. **Dev/Production API Key:** The API key provided by Radial for Development (Sandbox) or Production.
7. **Insight Medium Response Handle As:** Choose how the system will handle orders assessed as having a Medium risk of fraud.
  - Select *Low* to allow a customer to place an order with a Medium risk assessment.
  - Select *High* to prevent a customer from placing an order with a Medium risk assessment.



**Radial Risk Insight**

Environment [store view] Development ▼

Enabled [store view] Yes ▼

Radial Store ID [store view] MAGT2US

Dev API URL [store view] https://uat01-epapi-na.gsipartners.com  
Ex : https://example.com

Dev API Key [store view] 9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y  
Risk Insight Dev Api Key

Insight Medium Response Handle As [store view] High ▼  
Based on the seleted filed, medium response will handle by code

## Section 7: Configure Address Validation

To enable and configure Radial Address Validation:

1. Navigate to **Store > Configuration > Radial > Address Validation**.
2. In **Enable integration**, select **Yes** to enable the module and make address validation API calls in Checkout and the customer address section.

Values for the following fields will populate automatically when the module is installed. If necessary, enter the following default values or change the values to suit your store's requirements.

3. **Address Count:** This number of suggested addresses will be included in the Address Validation response.
4. In **Environment**, select *Sandbox* or *Production*.
5. **Dev/Production API URL:** The API URL provided by Radial for the Dev (Sandbox) or Production environment.

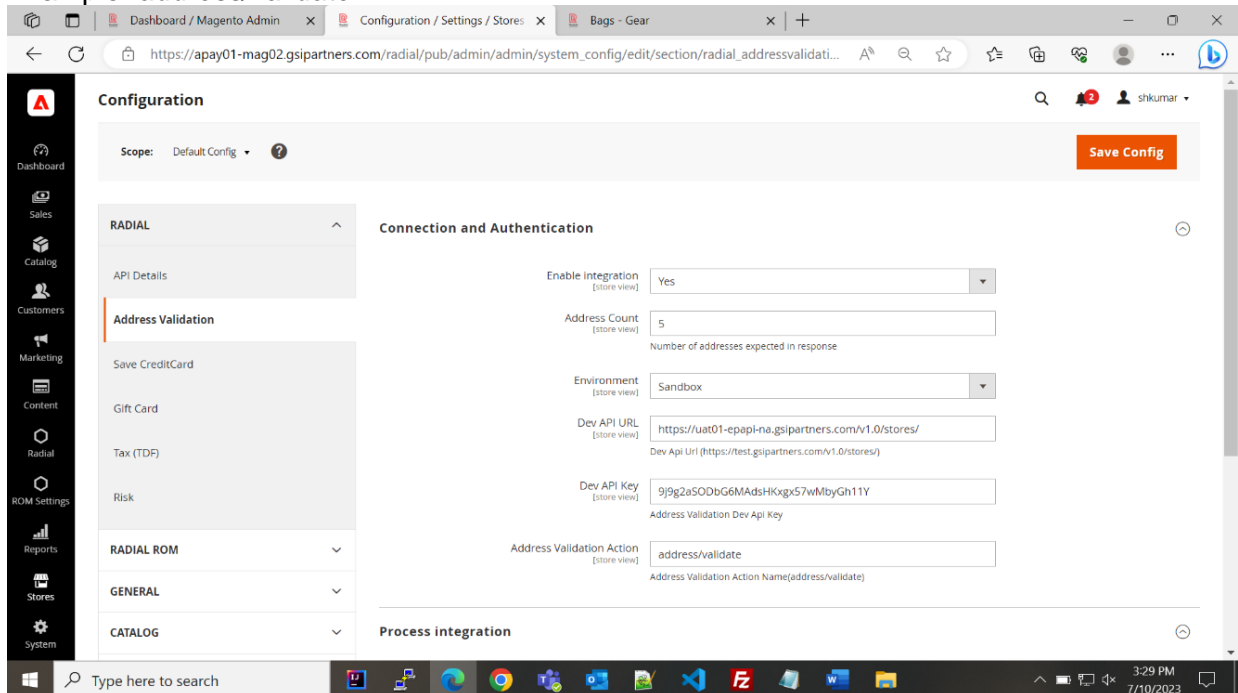
Example: *https://uat01-epapi-na.gsipartners.com*

6. **Dev/Production API Key:** The API Key provided by Radial for the Dev (Sandbox) or Production environment.

Example: *9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y*

7. **Address Validation Action:** The address validation action name.

Example: *address/validate*



The screenshot shows the Magento Admin configuration page for Radial Address Validation. The page is titled "Configuration" and has a "Scope" of "Default Config". The "Address Validation" section is selected in the left sidebar. The "Connection and Authentication" section is expanded, showing the following fields:

- Enable integration** (store view): Yes
- Address Count** (store view): 5 (Number of addresses expected in response)
- Environment** (store view): Sandbox
- Dev API URL** (store view): https://uat01-epapi-na.gsipartners.com/v1.0/stores/ (Dev Api Url (https://test.gsipartners.com/v1.0/stores/))
- Dev API Key** (store view): 9j9g2aSODbG6MAdsHKxgx57wMbyGh11Y (Address Validation Dev Api Key)
- Address Validation Action** (store view): address/validate (Address Validation Action Name(address/validate))

A "Save Config" button is visible in the top right corner of the configuration area.

## Section 8: Configure Webhook

If your store is a Radial Payments Solutions (PTF) client and is not using Radial Order Management (ROM), you must complete webhook configuration as described in this section.

If your store uses Radial Order Management (ROM), you do not need to complete this configuration.

To configure webhook for Radial, set up the following items in your webhook environment. In each case, replace *exampledomain.com* with the base URL for your store's Magento instance.

- **Settlement:** <https://exampledomain.com/radialpayment/webhook/settlement/>  
The Settlement webhook provides the response for Funds Capture and collection after the Payment Settlement call is made.
- **Authcancel:** <https://exampledomain.com/radialpayment/webhook/authcancel/>  
The Authcancel webhook provides the response for Authorization cancels that happen when the whole order is cancelled either due to Fraud assessment or from the warehouse.
- **RiskAssessment:** <https://exampledomain.com/risk/webhook/riskassessmentlist/>  
The RiskAssessment webhook provides the Risk response from Radial Fraud. A Risk response value for an order could be APPROVE or CANCEL or SUSPEND.
- **AsyncConfirmFunds:**  
<http://exampledomain.com/radialbanktransfer/webhook/asyncconfirmfunds/>  
The AsyncConfirmFunds webhook is used for Bank Transfer and similar tenders that do not pre-capture.