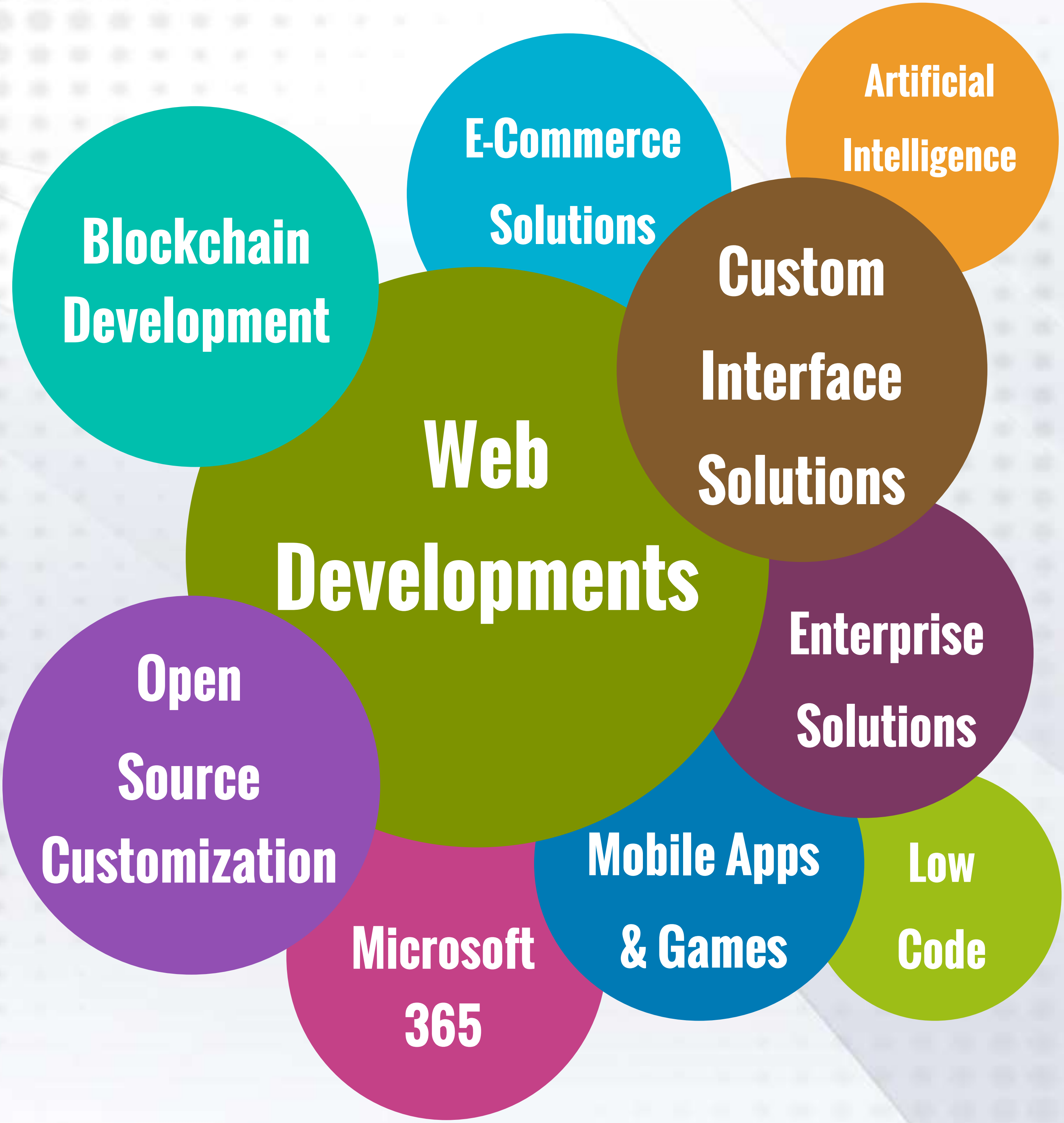
DART INNOVATIONS

ONE STOP IT SOLUTIONS

# One Shop for Web and Mobile Requirements



Blockchain Development

E-Commerce Solutions

Artificial Intelligence

Custom Interface Solutions

Web Developments

Enterprise Solutions

Open Source Customization

Microsoft 365

Mobile Apps & Games

Low Code

**User Guide**

**Product Key Delivery**

**Magneto 2.x Extension**

## Overview

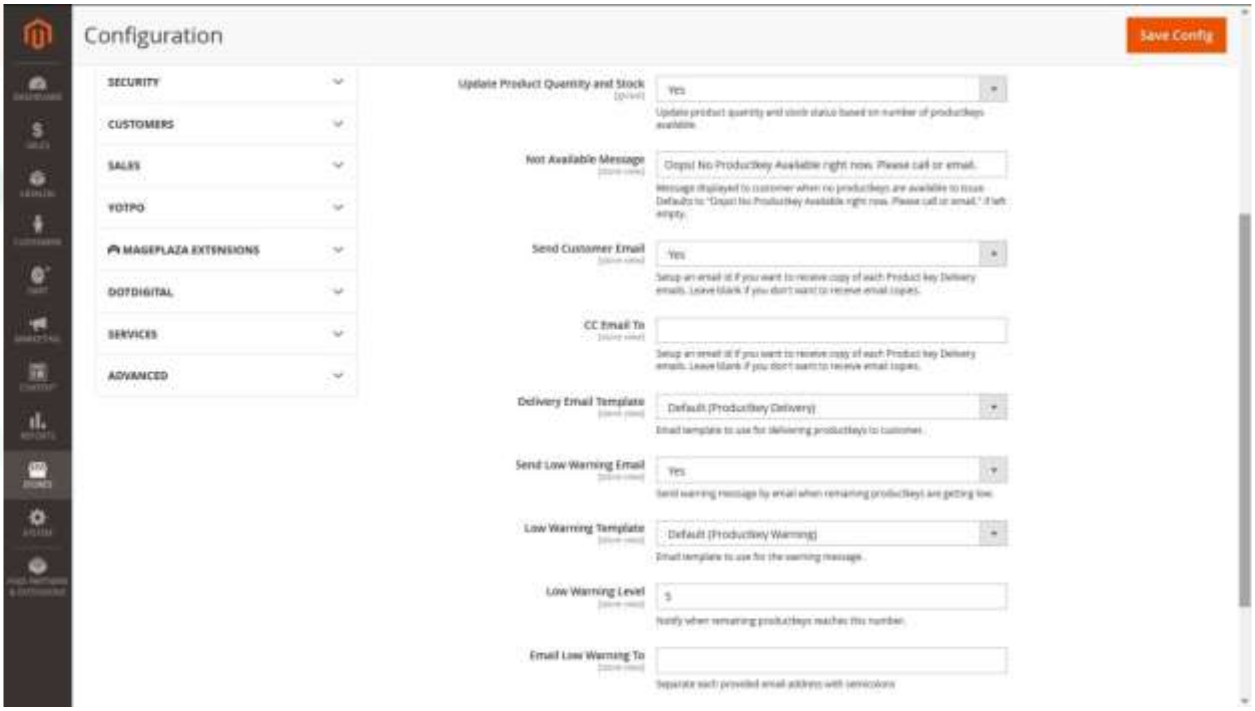
Thank you for purchasing Dart's Product Key Delivery plugin for Magento 2. This guide will help you getting started with your Product Key Delivery feature quick.

## Installation

- We offer only manual installation. Here are the simple steps to get it done.
- Download the zip package and extract it.
- Upload the folder **Dart** to the package to **/app/code/** directory.
- Issue the command ***php bin/magento setup:upgrade*** from your terminal.

## General Configuration

Click **Dart** then **Configuration** from the side bar. You will land into configuration page.



Update Product Quantity and Stock [global]: Yes
Update product quantity and stock status based on number of productkeys available.

Not Available Message [store view]: Oops! No ProductKey Available right now. Please call or email.
Message displayed to customer when no productkeys are available to issue. Defaults to "Oops! No ProductKey Available right now. Please call or email." if left empty.

Send Customer Email [store view]: Yes
Setup an email id if you want to receive copy of each Product key Delivery emails. Leave blank if you don't want to receive email copies.

CC Email To [store view]:
Setup an email id if you want to receive copy of each Product key Delivery emails. Leave blank if you don't want to receive email copies.

Delivery Email Template [store view]: Default (ProductKey Delivery)
Email template to use for delivering productkeys to customer.

Send Low Warning Email [store view]: Yes
Send warning message by email when remaining productkeys are getting low.

Low Warning Template [store view]: Default (ProductKey Warning)
Email template to use for the warning message.

Low Warning Level [store view]: 5
Notify when remaining productkeys reaches this number.

Email Low Warning To [store view]:
Separate each provided email address with semicolons

**Issue When Invoiced:** Yes/No (Setting Yes will send the key upon invoice)

**Product Key Type:** Setting a value here will be sent as title of key (Ex: Product Key, License Code etc)

**Update Product Quantity and Stock:** The product quantity will be updated based on the Key pool quantity if it is set to **Yes**.

**Not Available Message:** The message here will appear to user after successful checkout if a key is not available for that product.

**Send Customer Email:** Yes/No (Yes, will deliver key by email)

**CC Email To:** If you want each copy of product keys email you may place an admin email here. The email to customers will be sent to this email id as BCC. You can leave it blank if you don't need an email copy.

**Delivery Email Template:** You can select an email template here to use for product key delivery.

**Send Low warning Email**: Yes/No (Yes will send administrator an email saying the key pool is low on stock)

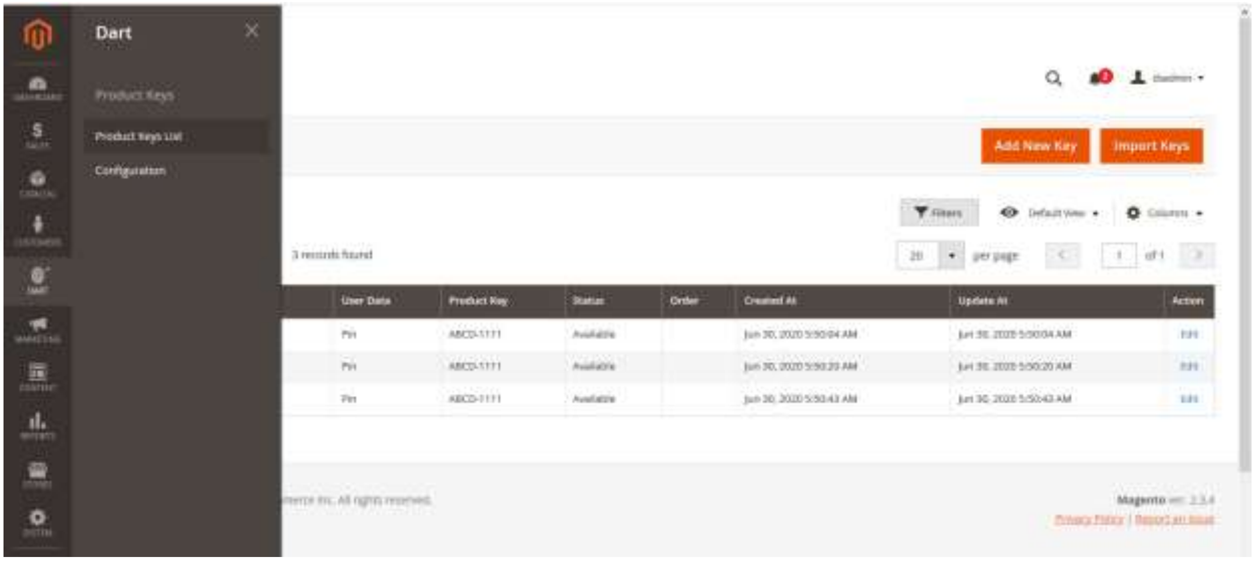
**Low Warning Template:** The email template that will be used to send when the key is lower than the warning level we set.

**Low Warning Level:** Set the value here so when the key pool is lower than this value will consider as lower stock and the notification will be sent to administrator. This prevents you from key pools going out of stock.

**Email Low Warning To:** Set an administrator email here which will receive your key pool low warning notifications.

## Keys List

Click **Dart** then **Product Keys List** from the side bar. You will see the grid with existing keys list.



3 records found

| User Data | Product Key | Status | Order | Created At | Update At | Action |
|---|---|---|---|---|---|---|
| Pin | ABCD-1111 | Available | | Jun 30, 2020 5:50:04 AM | Jun 30, 2020 5:50:04 AM | Edit |
| Pin | ABCD-1111 | Available | | Jun 30, 2020 5:50:20 AM | Jun 30, 2020 5:50:20 AM | Edit |
| Pin | ABCD-1111 | Available | | Jun 30, 2020 5:50:43 AM | Jun 30, 2020 5:50:43 AM | Edit |

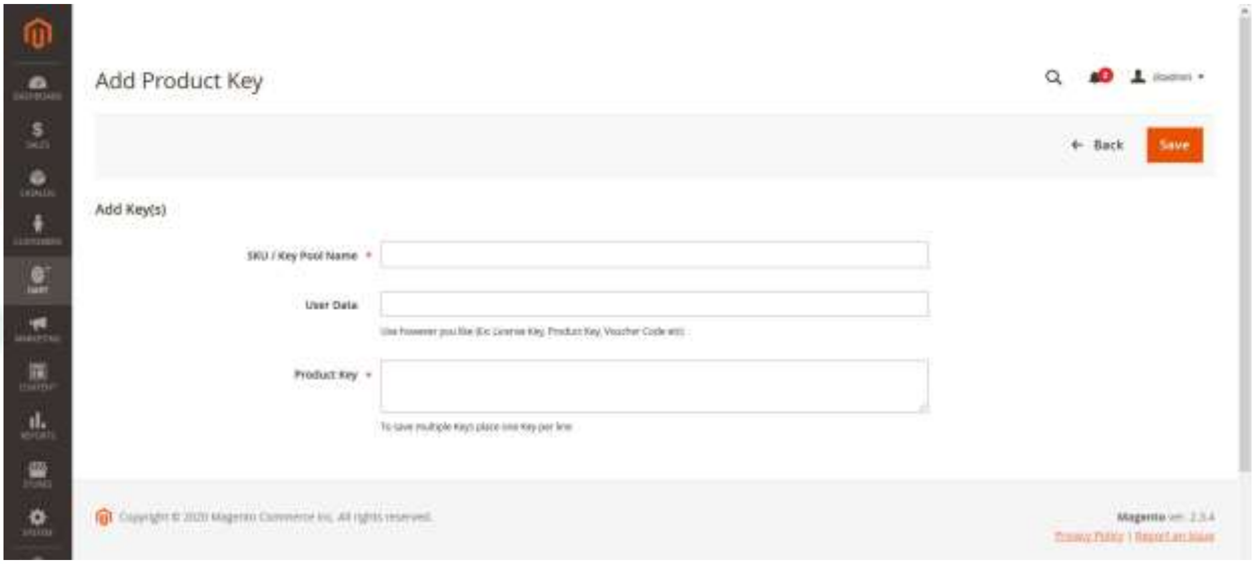
merce Inc. All rights reserved.

**Magento** ver. 2.3.4
Privacy Policy | Report an Issue

You can see the status as used which are meant as used with an order. You can also see the order id on grid. You can edit as well delete the keys on this grid.

Now click **Add New Key**, you will land into a page where you can enter new key(s).

**Add New Key:**



Add Product Key

Add Key(s)

SKU / Key Pool Name *

User Data

Use however you like (Ex: License Key, Product Key, Voucher Code etc) .

Product Key *

To save multiple Keys place one Key per line

Copyright © 2020 Magento Commerce Inc. All rights reserved.

Magento ver. 2.3.4
Privacy Policy | Report an Issue

Place a Key Pool Name or enter the SKU of the product you want to assign the key. Entering SKU will directly assign the key to that product where if you enter a pool name you have to map it on the product page.

You can enter single key or multiple keys by entering into new lines.

The field User Data is not mandatory but this is useful to use with email templates. This is to define the key type for example *License Key, Pin, Voucher Code* etc.

## Import Keys: (From plugin version 1.5.0)

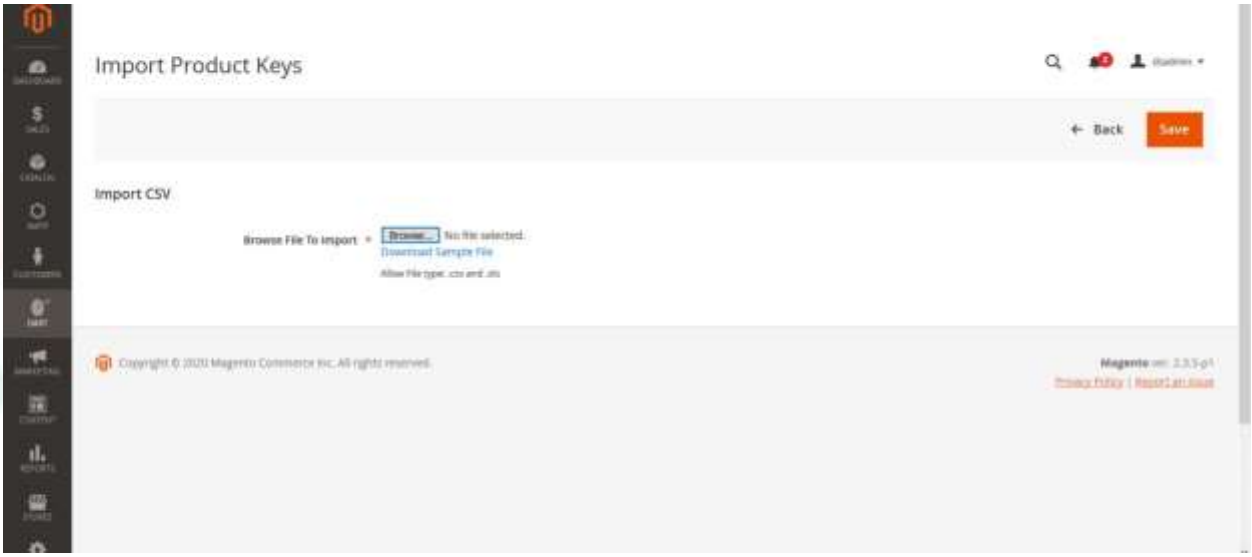
The Import Keys button allows you to land in new page where you can import CSV or Excel file with multiple keys and pools at one stretch.

A sample file is given which is the typical format of the CSV/excel that can be imported.

**type**: Give anything as you wish. Ex: (License Key, Product Key, Voucher Code etc)

**sku**: this column should have the Pool Name or SKU of the product that the key needs to mapped.

**product_key**: typically the Key.



Import Product Keys

Import CSV

Browse File To Import * Browse... No file selected.
Download Sample File
Allow File type: .csv and .xls

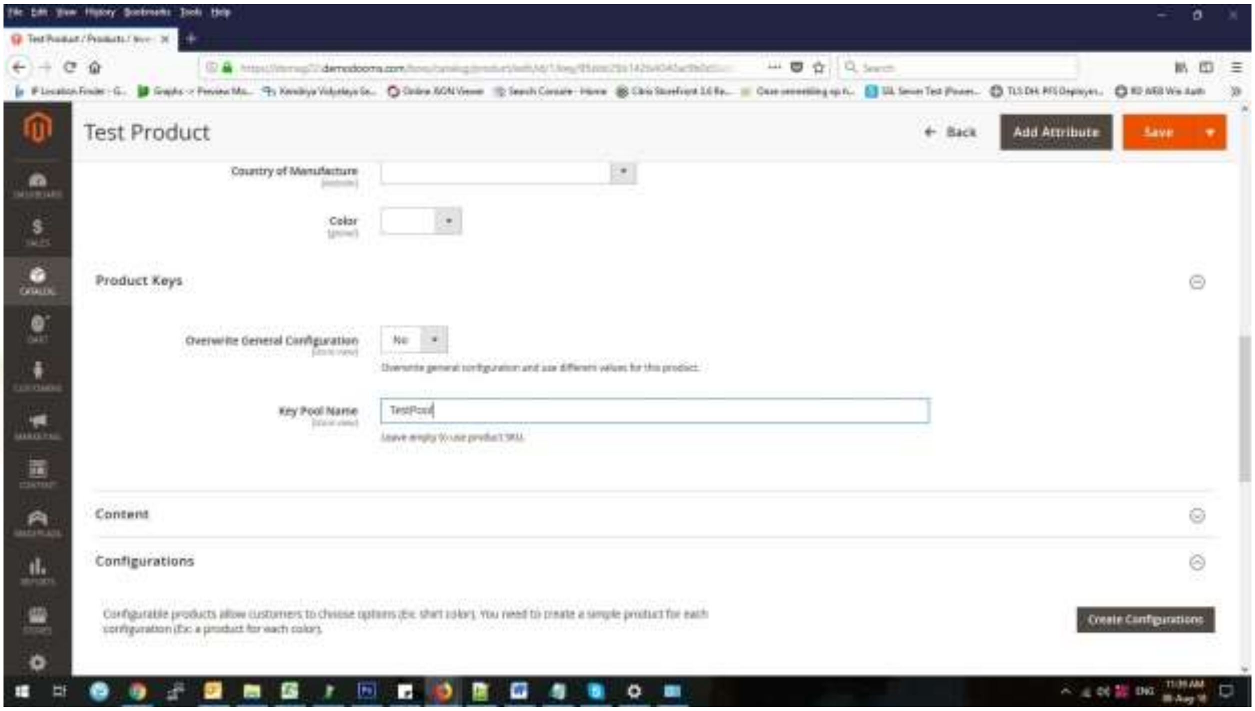
Back
Save
Copyright © 2020 Magento Commerce Inc. All rights reserved.
Magento ver. 2.3.5-p1
Privacy Policy | Report an Issue

## Mapping Key Pool to a product

You can map a key pool to single product or share against multiple products.

Go to **Catalog** >> **Products** and then edit a product you want to assign the key pool.

Scroll down and click on the Tab **Product Keys**



Test Product

Country of Manufacture [website]

Color [global]

Product Keys

Overwrite General Configuration [store view]: No
Overwrite general configuration and use different values for this product.

Key Pool Name [store view]: TestPool
Leave empty to use product SKU.

Content

Configurations

Configurable products allow customers to choose options (Ex: shirt color). You need to create a simple product for each configuration (Ex: a product for each color).

File Edit View History Bookmarks Tools Help
Test Product / Products / Inv...
demodooms.com
Search
Back
Add Attribute
Save
Create Configurations
11:39 AM
ENG

You can assign the Key Pool Name and the keys on that pool will be used for delivery. If you want to assign the same Key Pool to other product(s), repeat the steps.

The 'Overwrite General Configuration' will help you assign specific configuration for the product.

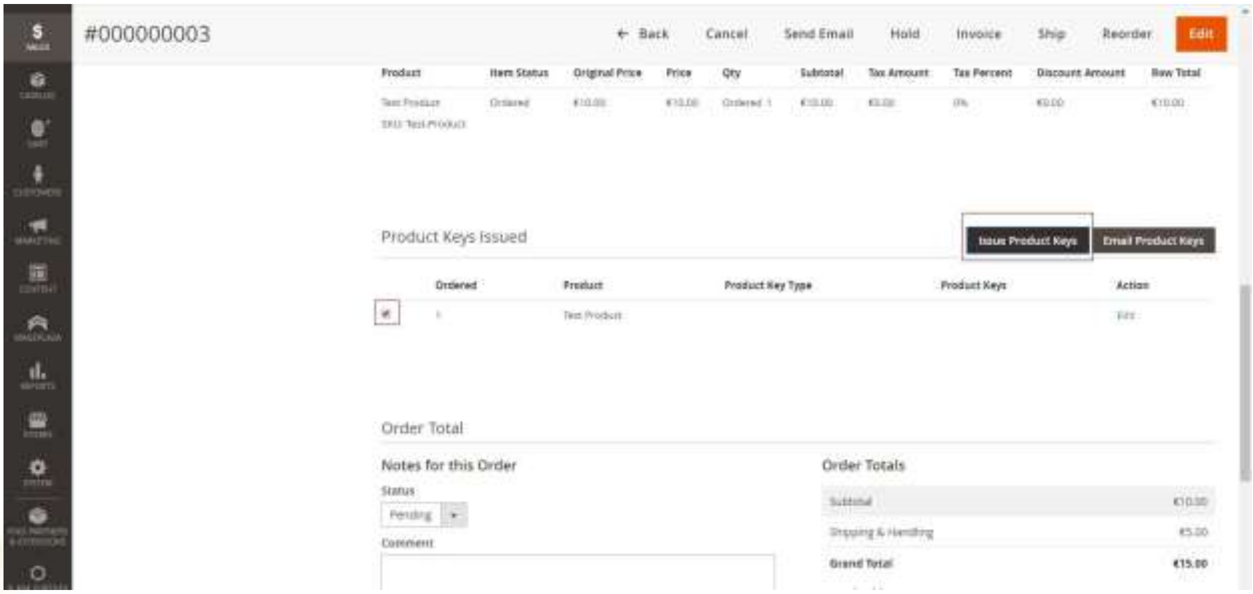
## Issue/Reissue the Keys Manually

There are occasions you may need to the issue the key from backend such cases like below.

- if you have set to send key after invoice as 'No'
- The order is made when your key pool is empty or you may like to edit.
- You want to edit the key and resend it.
- You may want to issue another key(s) on the same order.

Here is how you can do that.

Go and click on that particular order.



| Product | Item Status | Original Price | Price | Qty | Subtotal | Tax Amount | Tax Percent | Discount Amount | Row Total |
|---|---|---|---|---|---|---|---|---|---|
| Test Product<br>SKU: Test-Product | Ordered | €10.00 | €10.00 | Ordered 1 | €10.00 | €0.00 | 0% | €0.00 | €10.00 |

Product Keys Issued

| | Ordered | Product | Product Key Type | Product Keys | Action |
|---|---|---|---|---|---|
| ☑ | 1 | Test Product | | | Edit |

Order Total

Notes for this Order

Status: Pending

Comment

Order Totals

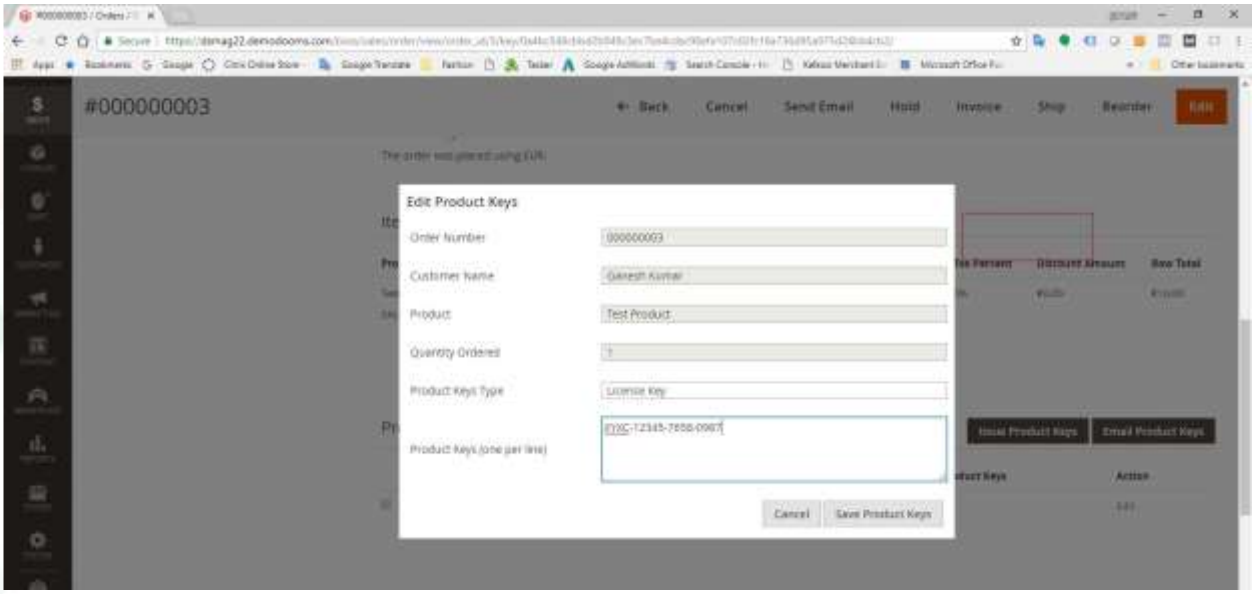
| | |
|---|---|
| Subtotal | €10.00 |
| Shipping & Handling | €5.00 |
| **Grand Total** | **€15.00** |

On the above page you can just select the product and click Issue Product Keys which will pick up a key from Key Pool and issue against this order. This is in assumption that you have already filled up the key pool for this product.

In other cases like,

- When you want to edit the existing key and reissue.
- Just like to send the Key directly from here, instead of adding to key pool.
- Or need to send another/add-on key(s)

Then Click Edit and you will get the below pop-up where you can add/edit key(s) and save.



Edit Product Keys

Order Number: 000000003

Customer Name: Ganesh Kumar

Product: Test Product

Quantity Ordered: 1

Product Keys Type: License Key

Product Keys (one per line): FYXC-12345-7658-0987

By saving will issue the Key(s) against that product. However if you want to email the key to user you need to select the product and click Email Product Keys button.

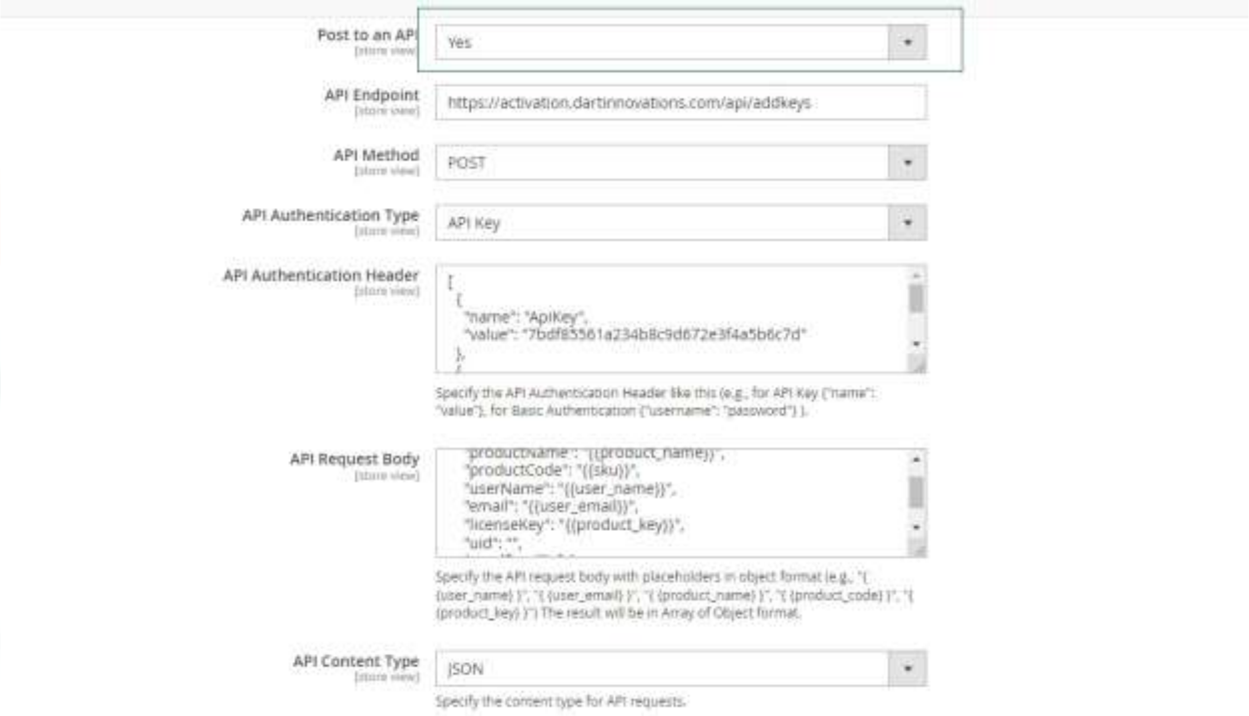
**Post to an API:** *(New - from version **2.0**)*

The new Post to an API allows you to integrate with external system using REST API calls. You can send the key with custom information to any external systems that has a proper REST API interface.

Go to **DART >> Product Keys >> Configuration**

Scroll down to **Post to an API** and select **Yes** to turn this feature on.

You will now see all options that can be used for integration with the 3rd party REST API. This is an advanced feature and obviously you just need to have bit of knowledge regarding your API the one which you are calling from here. Based on that, you have to populate the available fields to get it integrated with other REST API. Perhaps, it is very simple.



Post to an API [store view]: Yes

API Endpoint [store view]: https://activation.dartinnovations.com/api/addkeys

API Method [store view]: POST

API Authentication Type [store view]: API Key

API Authentication Header [store view]:

```
[
  {
    "name": "ApiKey",
    "value": "7bdf85561a234b8c9d672e3f4a5b6c7d"
  },
```

Specify the API Authentication Header like this (e.g., for API Key {"name": "value"}, for Basic Authentication {"username": "password"} ).

API Request Body [store view]:

```
    "productName": "{{product_name}}",
    "productCode": "{{sku}}",
    "userName": "{{user_name}}",
    "email": "{{user_email}}",
    "licenseKey": "{{product_key}}",
    "uid": "",
```

Specify the API request body with placeholders in object format (e.g., "{ {user_name} }", "{ {user_email} }", "{ {product_name} }", "{ {product_code} }", "{ {product_key} }") The result will be in Array of Object format.

API Content Type [store view]: JSON

Specify the content type for API requests.

**API Endpoint:** The full URL of the end point which you are intended to call.

**API Method:** you can use POST, PUT or GET method.

**API Authentication Method:** Basic Authentication, Bearer Token and API Key are the methods allowed.

**API Authentication Header:** Pass your raw header such like below example. You can include all your additional headers but should be properly formatted.

Example: We use to send our Product Keys to our activation portal which is basically a product called **License Web.** The **License Web** has an REST API interface which requires **API Key Pair.** In this case I pass my header like this,

```
 [
  {
    "name": "ApiKey",
    "value": "7bdf85561567b8c9d672e3fbnedf"
  },
  {
    "name": "ApiSecret",
    "value": "s1a234b8c9d672e3f4a5bftg"
  }
]
```

**API Request Body:** You can send any request body with proper format that your REST API end point requires.

Example, in my case our **License Web** portal requires below format.

```
{
   "productName": "{{product_name}}",
   "productCode": "{{sku}}",
   "userName": "{{user_name}}",
   "email": "{{user_email}}",
   "licenseKey": "{{product_key}}",
   "sendEmail": false
 }
```

**Short Codes:**
The below short codes are something that can be used to acquire environment variables during an order and used dynamically.

{{product_name}} The full name of the Product.

{{sku}} sku of the product *(License Web requires Product Code as an unique identifier where we use sku for that.)*

{{user_name}} User’s full name as per the order.

{{user_email}} Email of the ordering user.

{{product_key}} The product key code that has been delivered in this order.

You can also override the **Post to an API** settings on **product basis** as like the other settings of PKD.

--That’s all, Enjoy your Product Keys Delivery –

Documents and Resources: https://docs.dartinnovations.com/product-keys-delivery/

Support: https://products.dartinnovations.com/contact or reach us by live chat.