

CITRUS PRODUCT INTEGRATOR

User's Manual

How to install and configure the
Citrus-Magento Product Integrator,
to facilitate integration with the services
offered by Citrus International

Citrus Ad Extension for Magento 1.9.x

This library includes the files of the Citrus Ad extension.

Requirements

Magento 1.9.x

Installation

To install the extension on your Magento:

1. On your Magento admin panel, go to **System -> Magento Connect -> Magento Connect Manager**
2. Upload the `citrusad-[version_num].tgz` file and click **upload**.

Set up the Citrus Integrator

General settings

After the installation, go to **Magento admin panel -> System -> Configuration**, and click **CITRUS INTEGRATION -> General Setting**.

Enable the plugin in the Magento admin panel, input your Team Id and API Key from Citrus, and select the correct host.

NB: To retrieve your Team Id and API Key, log in to the Citrus client and go into the Integration Settings menu from the drop-down list in the top right corner.

Click **Save Config**.

When you have clicked **Save Config**, you will be able to sync products, customers, and orders, and enable ads and banners and add widgets.

Synchronization settings

Go to **Magento admin panel -> System -> Configuration**, and click on **CITRUS INTEGRATION -> Synchronization Option** on the left sidebar.

“Ads” (display ads) and “Banners” can be switched ON or OFF separately.

Enable synchronization for both “Customer/Orders” and “Products” and set sync modes as “Real-time”.

Click **Save Config**.

Click the following buttons: **“Add all customers to queue”**, **“Add all orders to queue”** and **“Add all products to queue”**.

Now that we have all the customers, orders, and products staged in the queue, go to **Magento admin panel -> Citrus -> Queue List**, select the items to submit, select **“sync”** and click **“Submit”**. This will sync the items with Citrus.

Troubleshooting

Conflicts arise when two (or more) modules rewrite the same class. In that case, the class will be overwritten by only one module, so the rest of the modules will not work properly. In some cases, this can have a fatal impact on your platform. Before you install this plugin, make sure that there will be no conflicts between in your installed plugins and our plugin. This can be done by checking the content of the <rewrite> in the "config.xml" files on all installed modules in your platform. Alternatively, you can use this tool module to help you detect the conflicts easily:

<https://marketplace.magento.com/alekseon-modules-conflict-detector.html>

If there is no conflict, no further configuration is necessary and the plugin will work as described.

If there are conflicts between the plugins, you must resolve them. You can use one of the following:

- * Merging - merge the code from one conflicting file into another and switch off the rewrite "config.xml" in one of them

or

- * Class inheritance - switch off the rewrite in one config.xml and then make the conflicting extension PHP file extend the other extension

or

- * both

Which of these you choose depends on the nature of the conflict.

In the Citrus plugin, the class `Mage_Catalog_Block_Product_List` is overwritten with the class `Citrus_Integration_Block_Product_List`.

```
<blocks>
  <catalog>
    <rewrite>
      <product_list>Citrus_Integration_Block_Product_List</product_list>
    </rewrite>
  </catalog>
</blocks>
```

The “Citrus_Integration_Block_Product_List” class overrides `_getProductCollection()` and adds two new methods: `getAdResponse()` and `sortByIndex()`.

If your plugins do not override the same methods, then you can use class inheritance and make the Citrus class extend yours (or vice versa) and then switch off the corresponding `<rewrite>`. If you want to use merging in this case, you must move the logic in the conflicting class to the one you want to merge to and switch off the corresponding `<rewrite>`.

If your plugins override the methods that Citrus requires, you might need to use merging with or without class inheritance. Make sure that the logic and functionality from both plugins is implemented correctly.

This requires understanding of the code and programming skills.

Furthermore, we implemented our click and impression functionalities in Javascript, which depends on the data in `list.phtml`.

If your plugins have modified this file, you must merge the logic of the Citrus plugin with the logic of your plugins in this file as well.

Logs

The requests log file is located at `/var/www/html/web/var/log/citrus.log`.

Licence

Apache Licence 2.0