Guide 2017_01

Suppliers & custom attributes

In this guide we provide instructions to:

- create a Storeden external application step by step;
- create an application that can manage suppliers;
- create custom attributes on Storeden products (Supplier, Supplier product price and Supplier product code) to associate suppliers to products;
- manage external data from your database in your application;
- publish your application.

Prerequisites:

- a Storeden store that you can handle
- an authorized developer account on https://developers.storeden.com
- a database. For this example we use MYSQL but you can adapt the code to your preferred database. Remember to provide your connection parameters on config.php file that you find in the package
- PHP Storeden SDK (provided in this example package)

Summary:

If you need assistance please write an e-mail to customercare@storeden.com or send a ticket from https://assistenza.storeden.com/

Create an application

To create an application, first you have to set the proper working space (that must be reached from internet).

Let’s begin with a root folder, and then generate this php files:

auth.php
embed.php
deauth.php
webhook.php

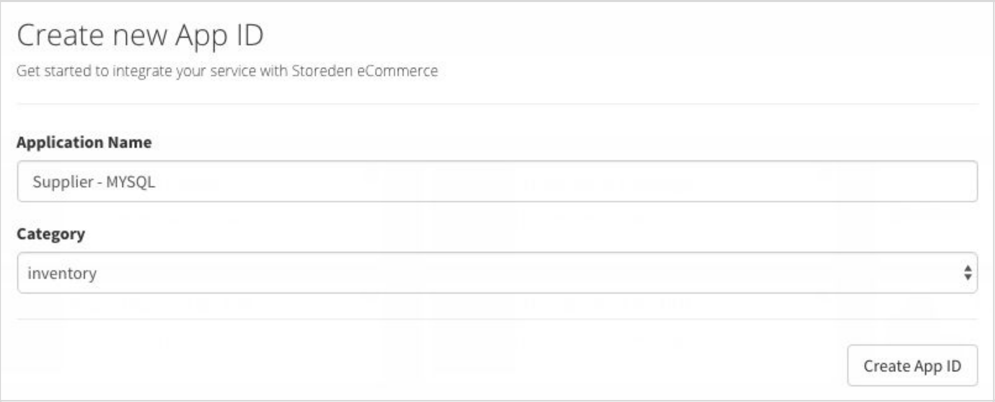
The auth.php file will be executed on app installation, here you need to save instance, key and secret of current installation. The deauth.php file will be executed on app uninstallation, here you need to revoke instance, key and secret of current installation. We will see this operations in details later.

In the embed.php file, you need to authenticate to the current app instance and after that, here you can make API calls. We will see this later.

The webhook.php file is also mandatory on app configuration, but for this example we don't need to examine this point.

Once you have set your app directory, let’s create an app in your developer account:
https://developers.storeden.com/apps/list

Press the green button “Create new App” on the right corner of the page.
Here you need to provide the name of your application and the category that the application belongs to.



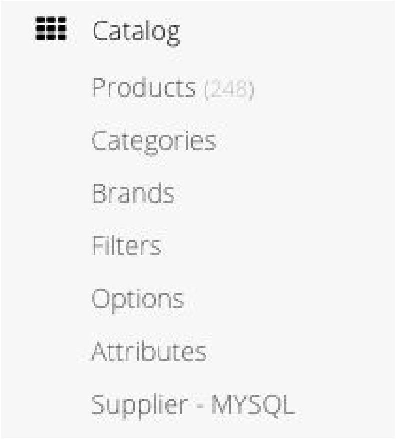
Create new App ID

Get started to integrate your service with Storeden eCommerce

Application Name: Supplier - MYSQL

Category: inventory

For our purpose, we provide “Supplier - MYSQL” as application name and “inventory” as application category.



Once the application is installed, the “inventory” category let the application’s link to appear under “Catalog” menu on the backoffice.

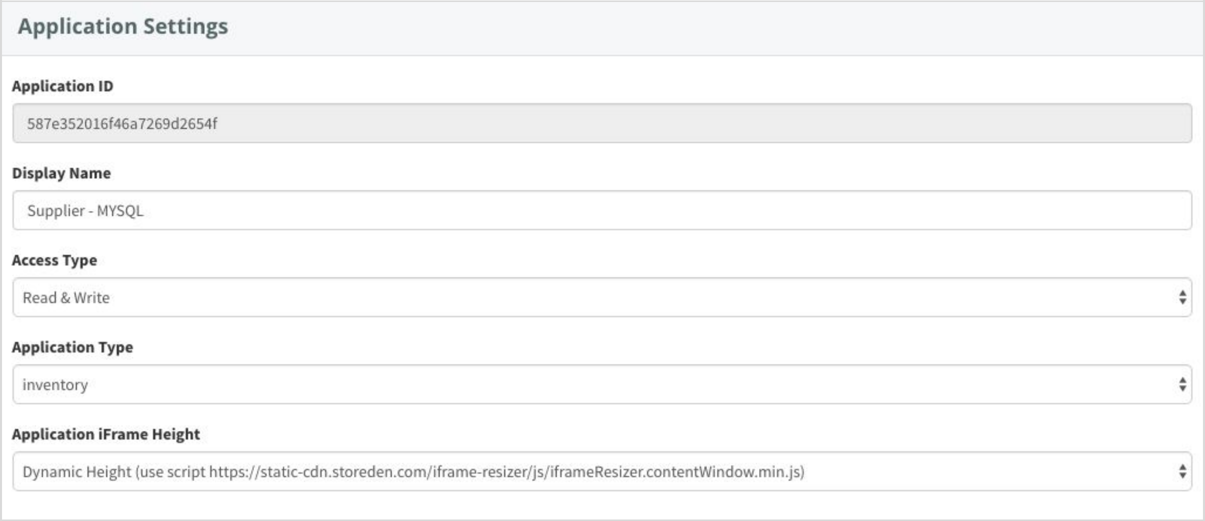
You can freely choose the category, but please remember that, for usability the best practice is to find the appropriate category for each application you develop, in accordance to their purpose.

Application settings

Now you need to provide all necessary settings to your application.

On **Application Access** choose “read/write” option and for **Application iFrame Height** select “Dynamic Height”. Please remember that you need to include this javascript file to your application to let iframe have a dynamic height:

https://static-cdn.storeden.com/iframe-resizer/js/iframeResizer.contentWindow.min.js



Application Settings

Application ID
587e352016f46a7269d2654f

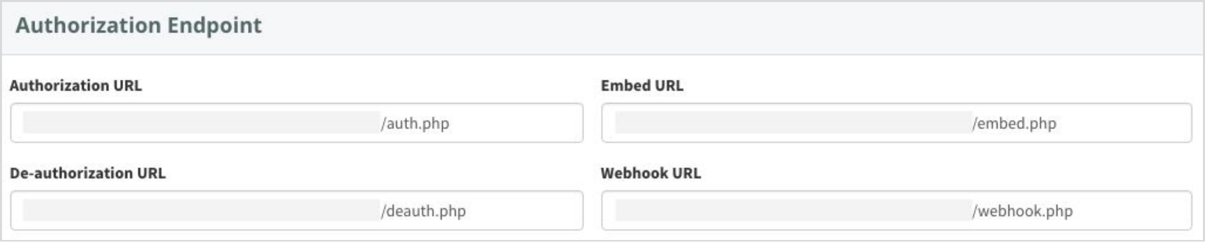
Display Name
Supplier - MYSQL

Access Type
Read & Write

Application Type
inventory

Application iFrame Height
Dynamic Height (use script https://static-cdn.storeden.com/iframe-resizer/js/iframeResizer.contentWindow.min.js)

After that, you need to provide the correct URLs of your previously created files.



Authorization Endpoint

Authorization URL
[redacted]/auth.php

Embed URL
[redacted]/embed.php

De-authorization URL
[redacted]/deauth.php

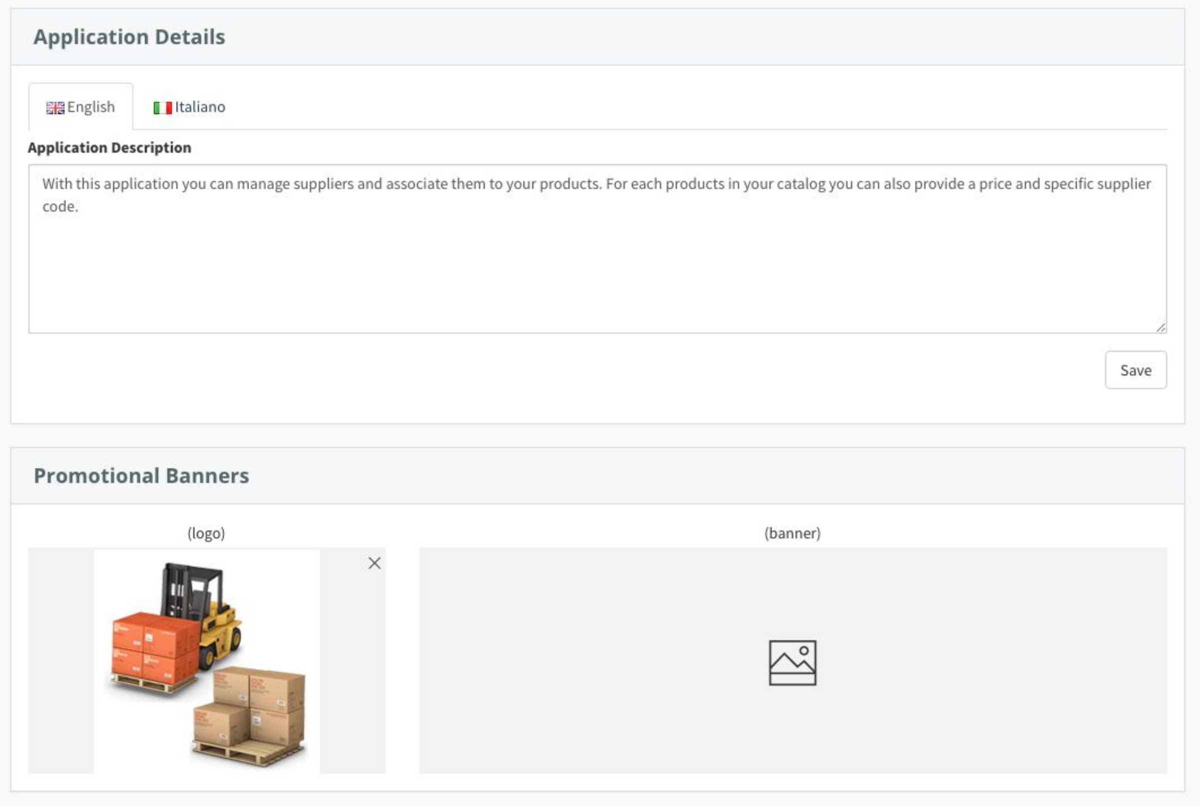
Webhook URL
[redacted]/webhook.php

At the end, you have to provide some customer care information: Privacy Policy URL, Terms URL, User Support email and a User Support URL.

Application details

On the “Details” page of your application, some descriptive information are required and also a logo image. The image must be square 300 x 300 px. You can find the image used for this example in the package.

Banner image and other screenshot are optional but recommended.

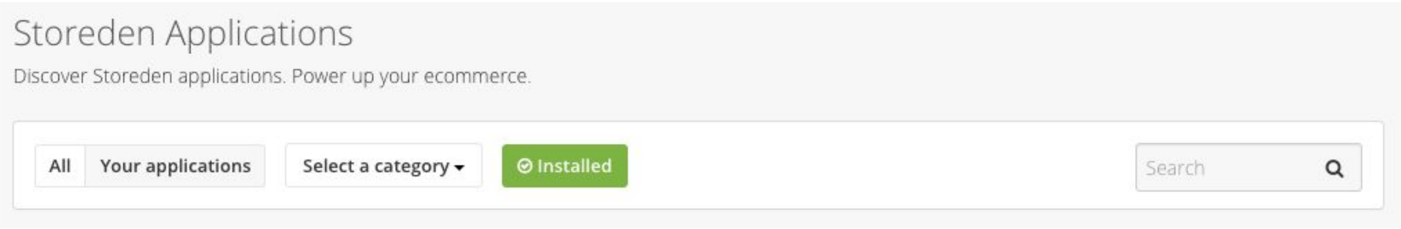


Application Details

Application Description

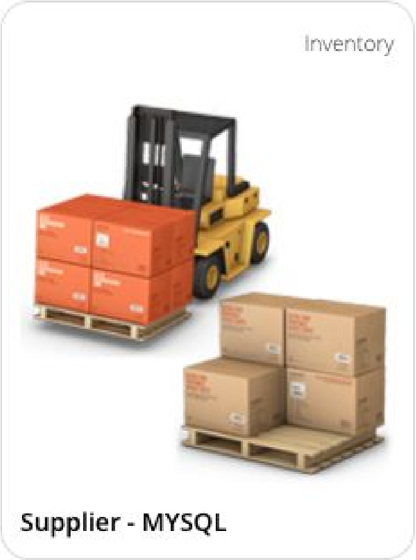
With this application you can manage suppliers and associate them to your products. For each products in your catalog you can also provide a price and specific supplier code.

Now you have your application, not published yet.
Go to your Storeden store backoffice, go to “Applications” section and click on “Your applications” in the toolbar.

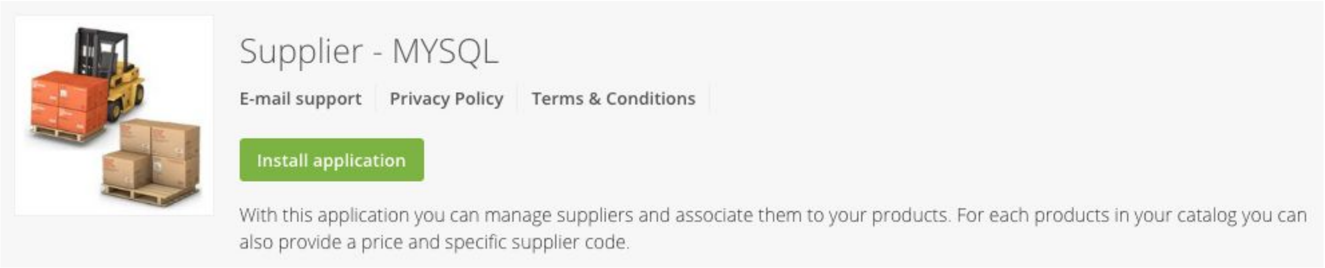

Storeden Applications

Discover Storeden applications. Power up your ecommerce.

You will find your application, but do not install it yet.



Application page will look like the screenshot below:



With this application you can manage suppliers and associate them to your products. For each products in your catalog you can also provide a price and specific supplier code.

Now let’s go deeper into the application development.

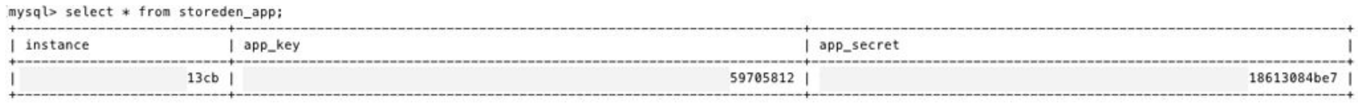
Application installation

Create “storeden_app” table in your database with these columns: instance (PK), app_key and app_secret.

After click on “Install” from application page, your auth.php file will receive in $_POST variable important parameters:

- instance
- key
- secret

You need to store this information in your local database to connect to Storeden API.
In this example, we save this information on “storeden_app” MYSQL table:



```
mysql> select * from storeden_app;
+---------------------------+--------------------------------------------------------+------------------------------------------------------+
| instance                  | app_key                                                | app_secret                                           |
+---------------------------+--------------------------------------------------------+------------------------------------------------------+
|                      13cb |                                               59705812 |                                         18613084be7 |
+---------------------------+--------------------------------------------------------+------------------------------------------------------+
```

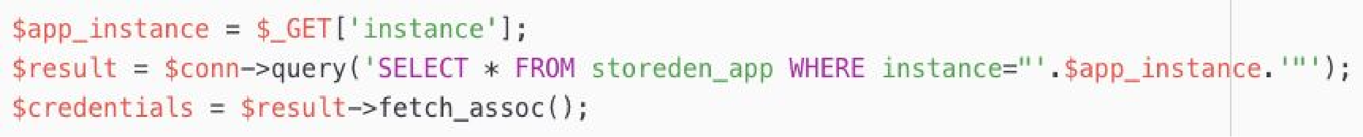
Application development

Now create “storeden_suppliers” table in your database with these columns: id (PK AI), label, email, phone, address, reference and picture.

Our application will:

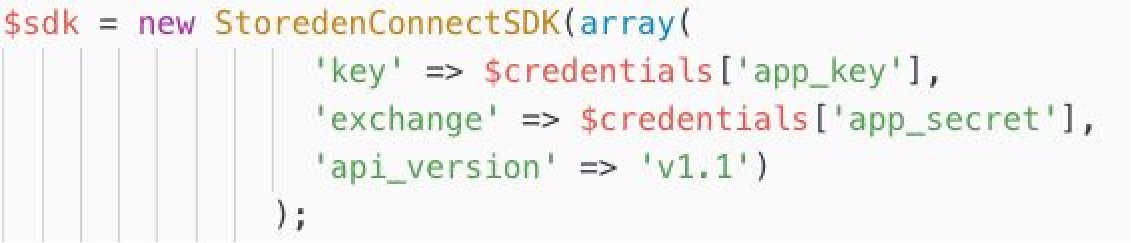
- create product custom attributes using API. The attributes will be created only if they don’t exist.
- list/add/update suppliers

We work now on the embed.php file.
First of all, we need to create a connection with Storeden SDK, in order to use Storeden API.
In the $_GET field, we find the app instance ID, so let’s use this ID to retrieve the “app_key” and “app_secret” codes from previously created “storeden_app” database table.



```
$app_instance = $_GET['instance'];
$result = $conn->query('SELECT * FROM storeden_app WHERE instance="'.$app_instance.'"');
$credentials = $result->fetch_assoc();
```

The `$credentials` array now must be used to create a valid SDK connection, like it is shown below:



```
$sdk = new StoredenConnectSDK(array(
                'key' => $credentials['app_key'],
                'exchange' => $credentials['app_secret'],
                'api_version' => 'v1.1')
            );
```

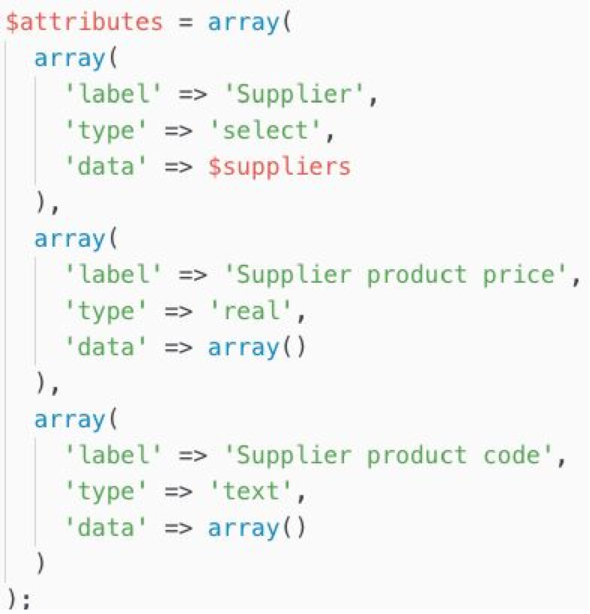
We have now a `$sdk` object that can be used to make API calls.
Here you find a list of Storeden provided API: https://developers.storeden.com/docs

For our purpose, we need to add some custom attributes in our application to use in store catalog.

- **Supplier** : this attribute will store the supplier name to associate to the products. This attribute will be a dropdown select value, because we will provide a list of supplier;
- **Supplier product price**: the price provided by the supplier;
- **Supplier product code**: the product internal code provided by the supplier;

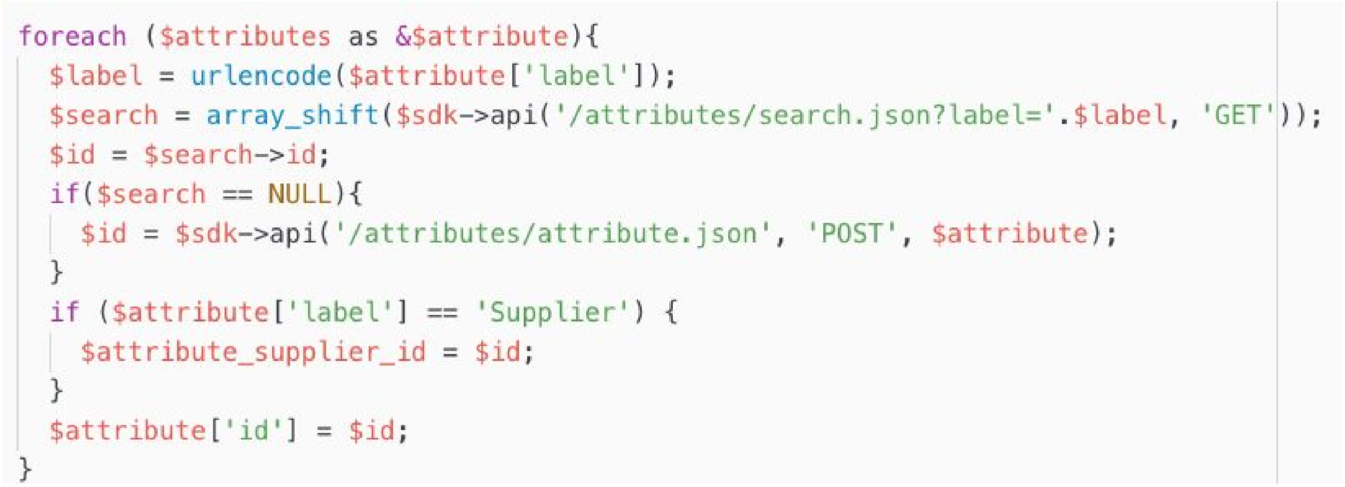
Please refer to this guide to provide proper attributes types:
https://developers.storeden.com/docs/attribute

Supposing that `$suppliers` is an array of suppliers name, the attributes definition will look like it is shown below:



```php
$attributes = array(
  array(
    'label' => 'Supplier',
    'type' => 'select',
    'data' => $suppliers
  ),
  array(
    'label' => 'Supplier product price',
    'type' => 'real',
    'data' => array()
  ),
  array(
    'label' => 'Supplier product code',
    'type' => 'text',
    'data' => array()
  )
);
```

Now we need to check if attributes already exists in the store and, if they don't exist, we need to create them. This check is recommended because the user can remove attributes from backoffice.



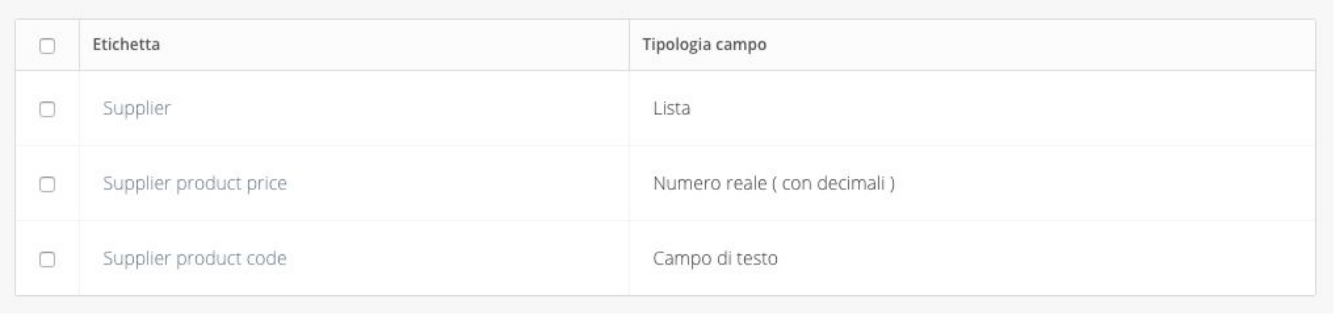
```php
foreach ($attributes as &$attribute){
  $label = urlencode($attribute['label']);
  $search = array_shift($sdk->api('/attributes/search.json?label='.$label, 'GET'));
  $id = $search->id;
  if($search == NULL){
    $id = $sdk->api('/attributes/attribute.json', 'POST', $attribute);
  }
  if ($attribute['label'] == 'Supplier') {
    $attribute_supplier_id = $id;
  }
  $attribute['id'] = $id;
}
```

The API call below is used to search an attribute by its label.
`$sdk->api('/attributes/search.json?label='.$label, 'GET')`

The following API call is used to insert a new attribute
`$sdk->api('/attributes/attribute.json', 'POST', $attribute)`

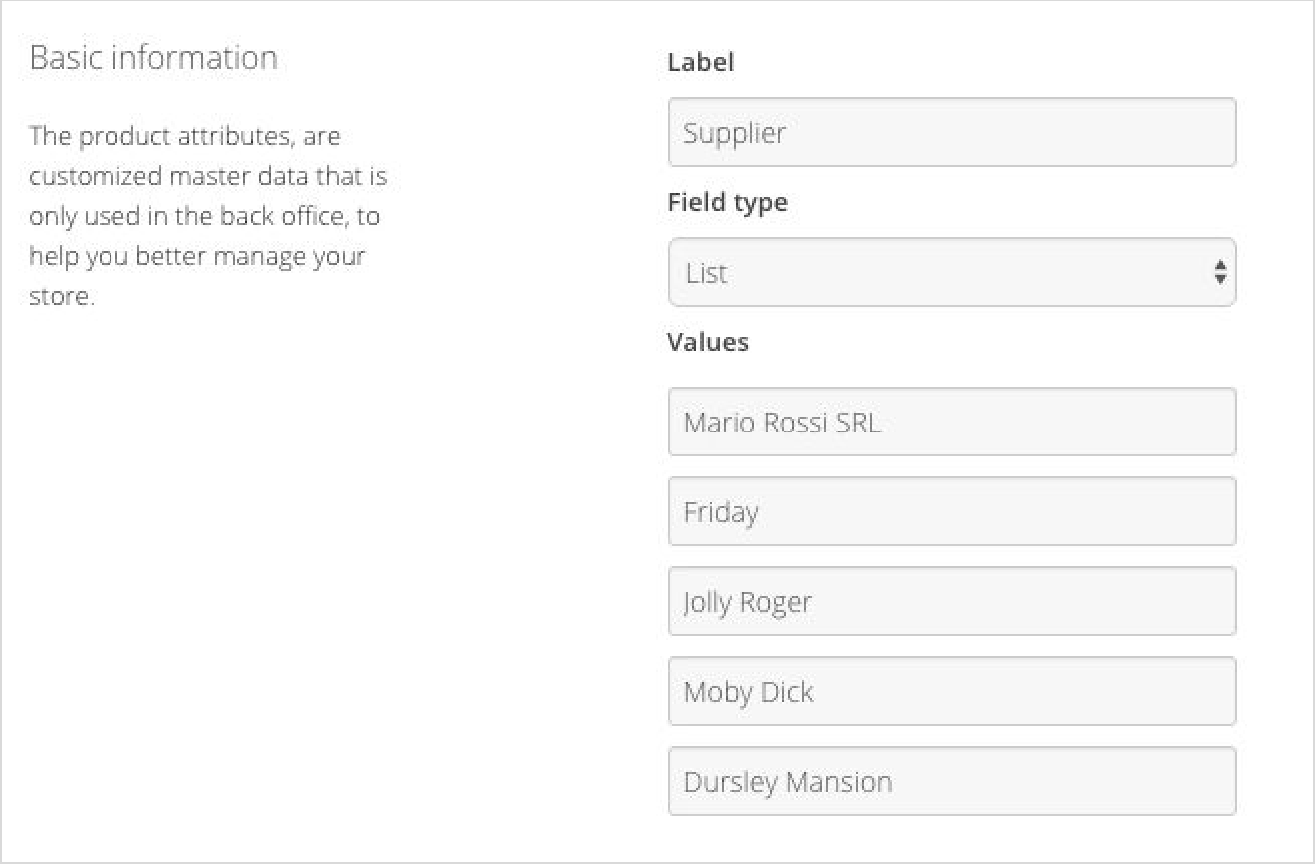
`$attribute_supplier_id` may be used to update the suppliers list stored in the attribute select, for example when you add a new supplier.

After that, in your backoffice (Catalog > Attributes) you will find that three attributes have been created:



□	Etichetta	Tipologia campo
□	Supplier	Lista
□	Supplier product price	Numero reale (con decimali)
□	Supplier product code	Campo di testo

Supposing you have some supplier in your database, the attribute list will look like you see hereunder.



Basic information

The product attributes, are customized master data that is only used in the back office, to help you better manage your store.

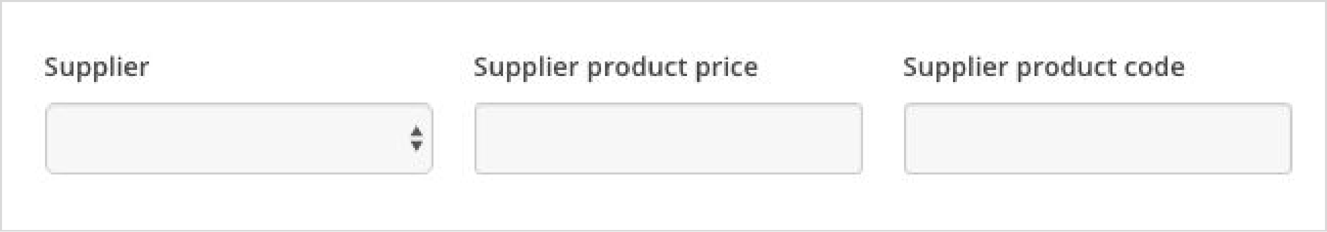
Label: Supplier

Field type: List

Values:
- Mario Rossi SRL
- Friday
- Jolly Roger
- Moby Dick
- Dursley Mansion

If the user remove the attribute or change its label, the application will recreate it once reloaded. But it will have a different ID so the products won't maintain the association.

In the product add/edit pages, now the user can insert and edit product information according to the newly created attributes.



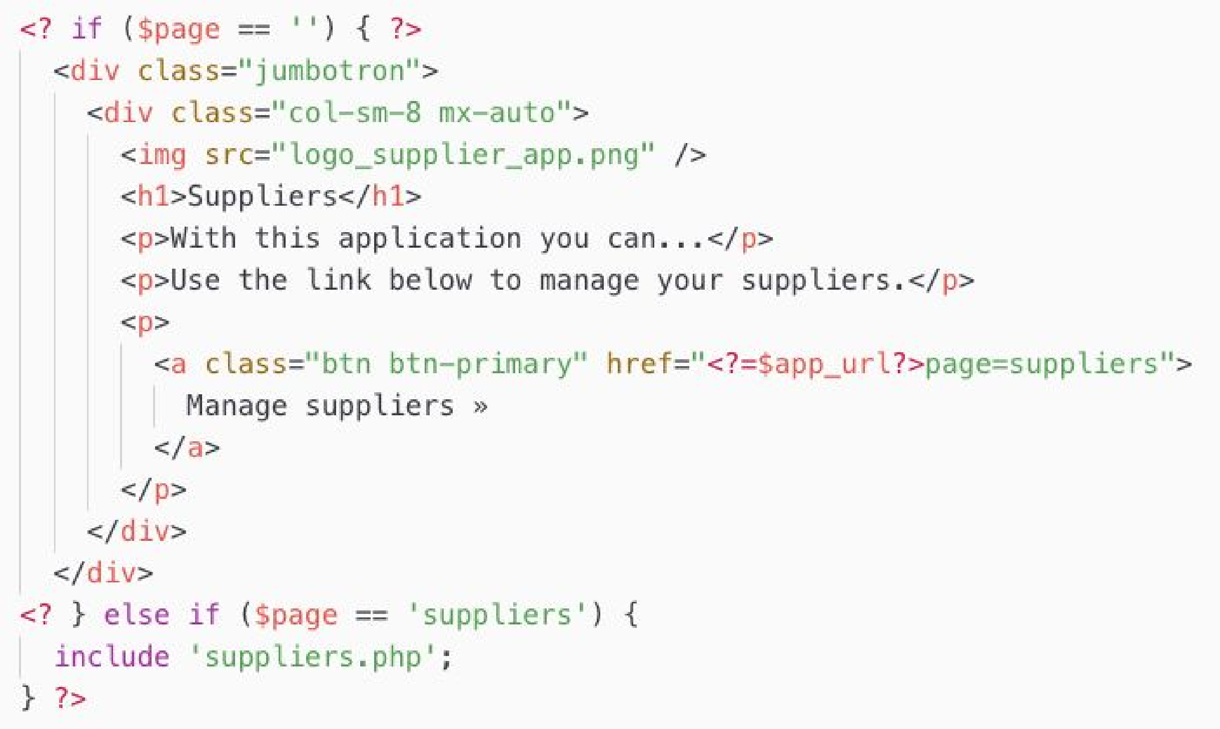
Supplier	Supplier product price	Supplier product code

After this operations, your application need to show some contents; for this app we create a “homepage” and a “detail page” but your interface can be as you want, in accordance to your application purpose.

Let’s create a new file layout.php and include it into embed.php. If you prefer, you can also continue to work on embed.php.

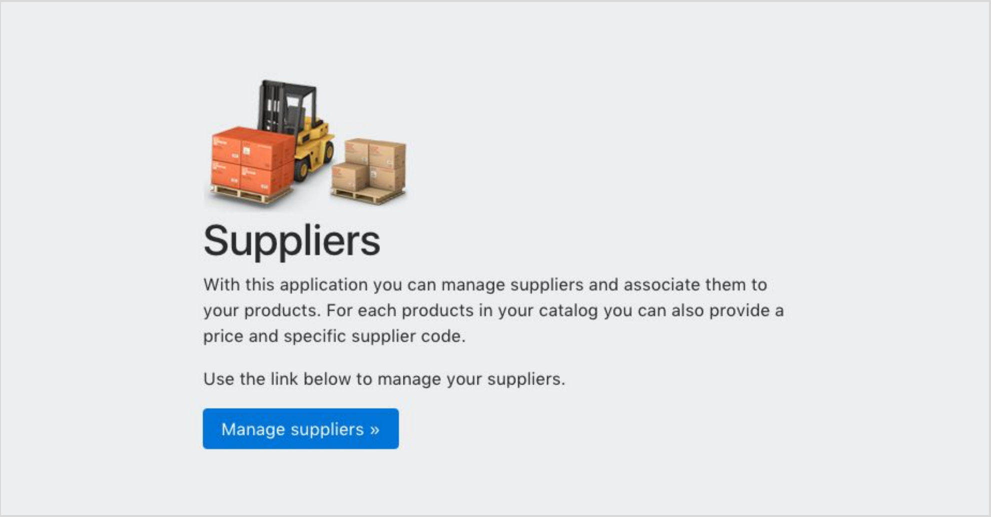
In layout.php create a html page, and here include css, javascript and every file you need. We used Bootstrap v. 4 alpha and Font Awesome to build the pages, but you can use the technique you prefer.

Supposing to have a $page variable that can handle routing, we distinguish the homepage from the suppliers page.



```
<? if ($page == '') { ?>
  <div class="jumbotron">
    <div class="col-sm-8 mx-auto">
      <img src="logo_supplier_app.png" />
      <h1>Suppliers</h1>
      <p>With this application you can...</p>
      <p>Use the link below to manage your suppliers.</p>
      <p>
        <a class="btn btn-primary" href="<?=$app_url?>page=suppliers">
          Manage suppliers »
        </a>
      </p>
    </div>
  </div>
<? } else if ($page == 'suppliers') {
  include 'suppliers.php';
} ?>
```

This code will produce the following user interface in your application page:



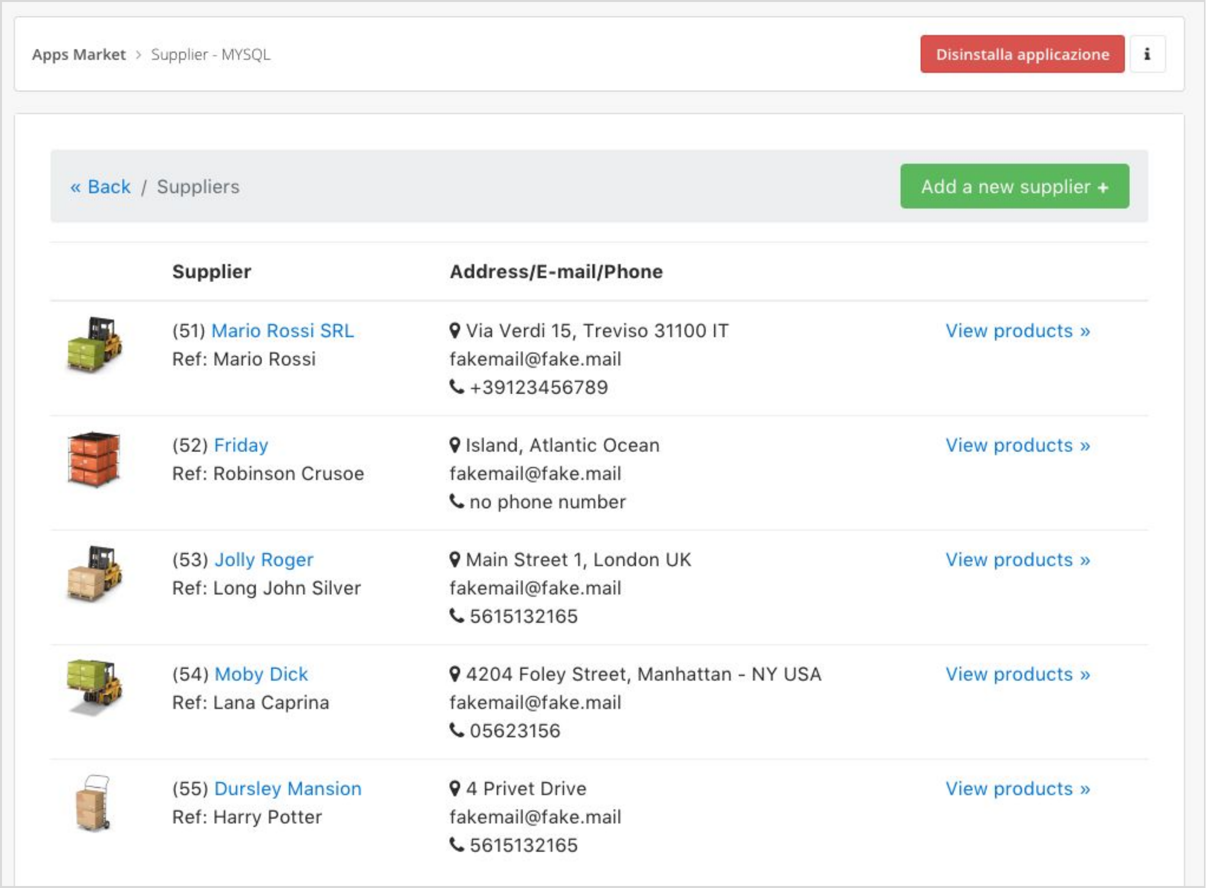
Suppliers

With this application you can manage suppliers and associate them to your products. For each products in your catalog you can also provide a price and specific supplier code.

Use the link below to manage your suppliers.

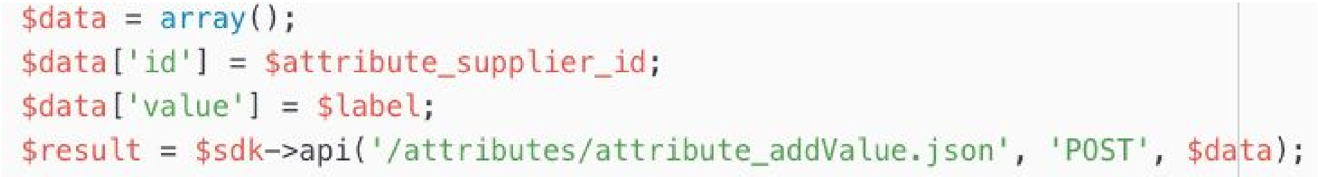
Let’s now create suppliers.php. This file will handle the list/add/edit processes of suppliers.

By clicking on “Manage suppliers” button, we expect a result like shown below. This is a simple table with an add form. The edit form is shown when click on supplier name.



	Supplier	Address/E-mail/Phone	
	(51) Mario Rossi SRL Ref: Mario Rossi	📍 Via Verdi 15, Treviso 31100 IT fakemail@fake.mail 📞 +39123456789	View products »
	(52) Friday Ref: Robinson Crusoe	📍 Island, Atlantic Ocean fakemail@fake.mail 📞 no phone number	View products »
	(53) Jolly Roger Ref: Long John Silver	📍 Main Street 1, London UK fakemail@fake.mail 📞 5615132165	View products »
	(54) Moby Dick Ref: Lana Caprina	📍 4204 Foley Street, Manhattan - NY USA fakemail@fake.mail 📞 05623156	View products »
	(55) Dursley Mansion Ref: Harry Potter	📍 4 Privet Drive fakemail@fake.mail 📞 5615132165	View products »

In your application, you can handle form action to add/edit suppliers on your database, the only important thing is to update the “Supplier” attribute when you add a new supplier, so the list in backoffice is always up to date. This can be done by making a specific API call:



```
$data = array();
$data['id'] = $attribute_supplier_id;
$data['value'] = $label;
$result = $sdk->api('/attributes/attribute_addValue.json', 'POST', $data);
```

The variable `$attribute_supplier_id` is used to update attribute lists. After a new supplier is inserted, then add a new value to select list of specified attributes.

`$attribute_supplier_id` is also used here to build the link to supplier’s products. The link will bring the user to Catalog > Products filtered by supplier.

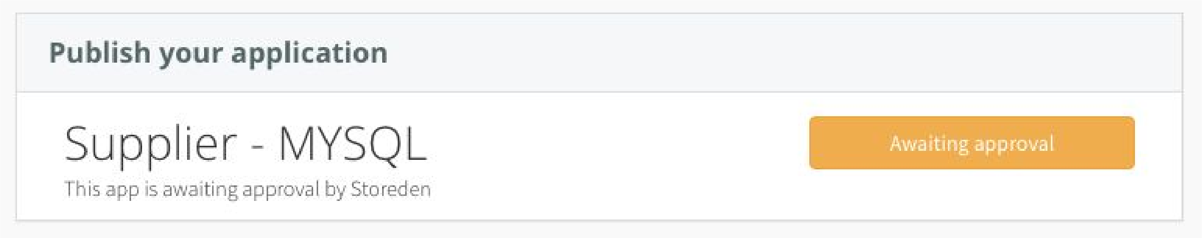
In the package provided, you will find the complete example.

Application uninstall

The uninstall of the application can be done in deauth.php file. Here you may need to remove app key and secret that you have stored for the current application instance (because they will not be needed anymore).

Publish application

In order to publish your application to the app market, you need to go to your developer account, go to “My apps” page, then click on your application. From here, select “Publish” section in the left column. Click on “Public” button and then wait for Storeden administration approval.

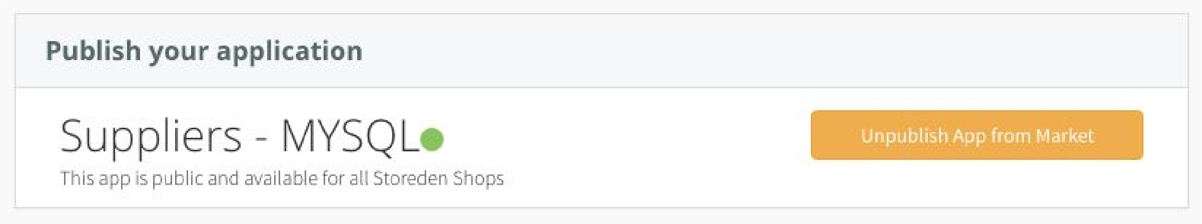


Publish your application

Supplier - MYSQL

This app is awaiting approval by Storeden

After administration approval, your application will be visible into the Apps Market and can be installed by every store that want it.



Publish your application

Suppliers - MYSQL●

This app is public and available for all Storeden Shops