Localization

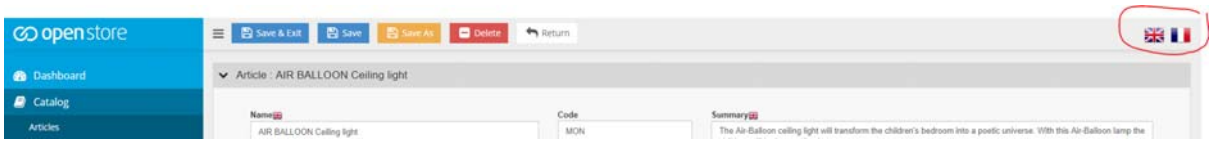
Introduction

OpenStore uses DB records to be fully localizable. But there is still the problem of displaying and updating the correct language.

Admin Panel

Most situations that need an edit language updated are saving data from the OpenStore Admin Panel, but there are other areas, like the checkout where language may be required.

In the Admin Panel the edit language is controlled by the flags on the top right of the display. Please note this is the language you are editing, the “uilang” and the language for the labels is taken from DNN display language when you enter the Admin Panel.



Razor Templating

OpenStore deals with language editing by using a number of hidden fields in the razor template, which define to the server code what language is being view, edited or changed.

It is very important the server understands which language it is editing and displaying, in OpenStore the updates are often done using Ajax and hence the current context and language may not be available in code. We therefore pass this language data via hidden fields in the html.

In the past there were no standards for the name of these fields and so you may find other names in the templates. However now you should try and implement a standard naming convention.

```
<input id="uilang" type="hidden" value="@Model.GetUrlParam("language")" />
```

"uilang" is used to set the language of the UI interface, this should match the DNN language, but in some calls to ajax the DNN language is lost, hence the need to use this field in the template to ensure ajax calls use the correct UI language when rendering templates. In the example we have used the language param from the url, in most cases this will be correct.

```
<input id="editlang" type="hidden" value="@Model.Lang" />
```

"editlang" is the data editing language that is being updated. This is taken from the razor model data “Lang” field, this should match the data being displayed for edit.

```
<input id="nextlang" type="hidden" value="@Model.Lang" />
```

"nextlang" is the store used on language change to identify the next language that should be read from the Database. This is used for ajax calls when the current edit language must be passed back to the server to update data in the correct language, but the next language to be read is different ("nextlang"). The nextlang is usually copied to the "editlang" record after the update is done, then a read with the next langauge (in "editlang") will be made. This adds a level of complexity, because we are doing a save and display on edit language change. An easier way to deal with the edit language is to hide the edit language flags once a change to the data has been made, this way it will force the user to save, making the edit flags appear, before changing languages.