

Technical Documentation

**Document ID : 5461**

**5456**

**Document Version : 1.0**

**Author : Prathvi Shetty**

**Niharika M Acharya**

**Creating Module :**

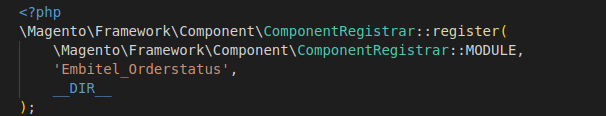
In this project we have created a module named “Orderstatus” with a “Embitel” namespace.

The path of the Module is:

**app/code/Embitel/Orderstatus**

**Steps followed while creating Module:**

* Created registration.php for registering the module.

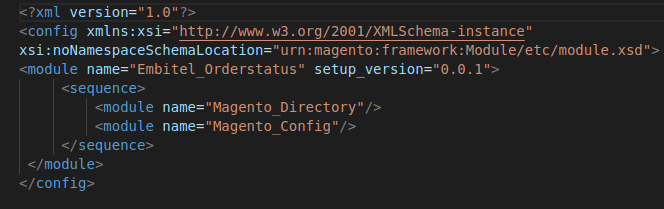


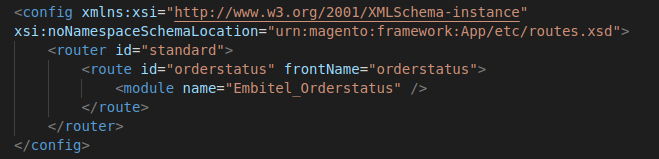
* Created module.xml inside etc folder.

**etc/module.xml**

Then created routes.xml to create the route in the module.

**app/code/Embitel/Orderstatus/etc/frontend/routes.xml.**

****

****

**Creating Admintheme:**

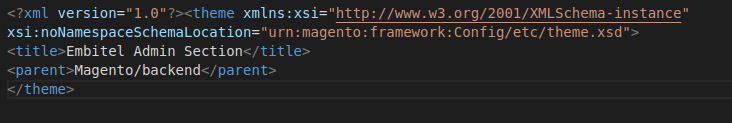
* Create Theme Directory

In the app/design/adminhtml directory, added the new directory: **Embitel/Admintheme**

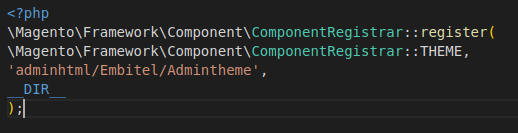
* Create theme.xml file

The path is-

**app/design/adminhtml /Embitel/Admintheme/theme.xml**



* Then created the registration.php file



### Create web folder to save media file

**app/design/adminhtml/Embitel/Admintheme/web/Images/logo.png**

****

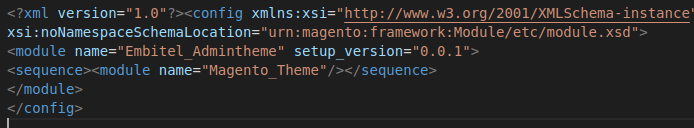
* Create Module to Declare Theme

To declare theme we need module which is already created named as “Orderstatus”.

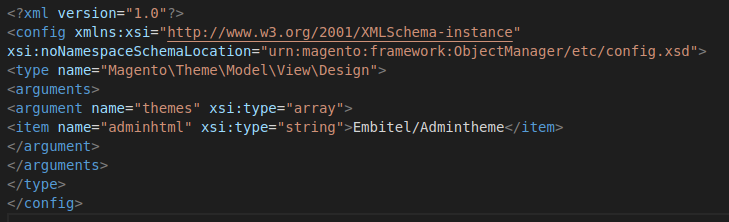
* Then, added the folder “etc/” and created module.xml file and di.xml file.

The path is-

**app/code/Embitel/Admintheme/etc/module.xml**

****

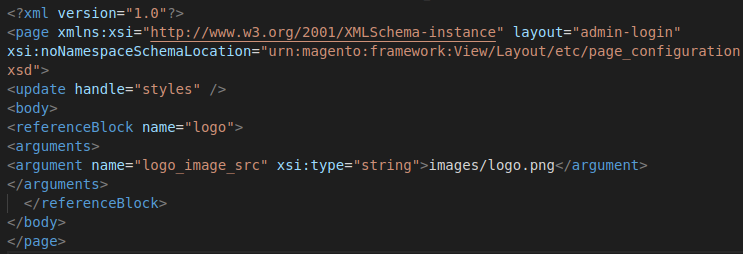
**app/code/Embitel/Admintheme/etc/di.xml**



* Then created “view/” folder to push all media files and layout

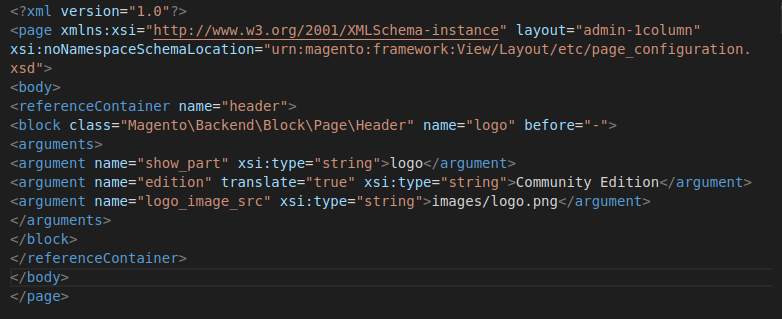
The path is-

**app/code/Embitel/Admintheme/view/adminhtml/layout/admin\_login.xml**



* Then we created default.xml file.
* The path is-

**app/code/Embitel/Admintheme/view/adminhtml/layout/default.xml**

****

* **Changing Order status from “COMPLETE” to “DISPATCHED”**

This task we have done in admin page .The path followed is

**stores/Order Status/Create New Status**

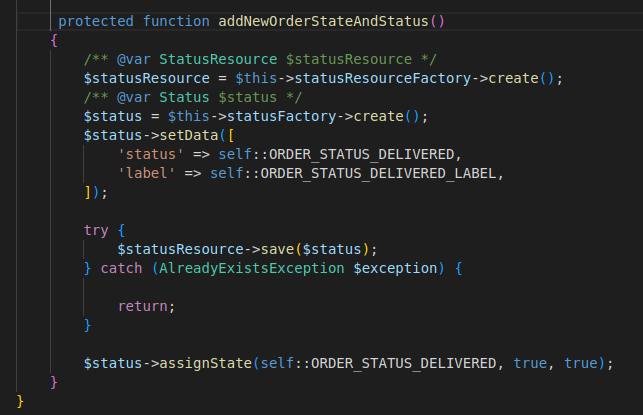
Then we have to alter the status Label from “COMPLETE” to “DISPATCHED”.

The changes will reflect in order view page where instead of “COMPLETE” ,it

will show as “Dispatched”.

* **Create new order state and status**

To create new order state and status,first of all we need to create module under namespace which we already created. Inside module we have created Setup folder and placed InstallData.php file inside it.

****

* **Create a button in order view when order status is “DISPATCHED”.The button should visible only when the order status is updated as “DISPATCHED” and once the button function is completed , it should disappear.**

Steps followed are:

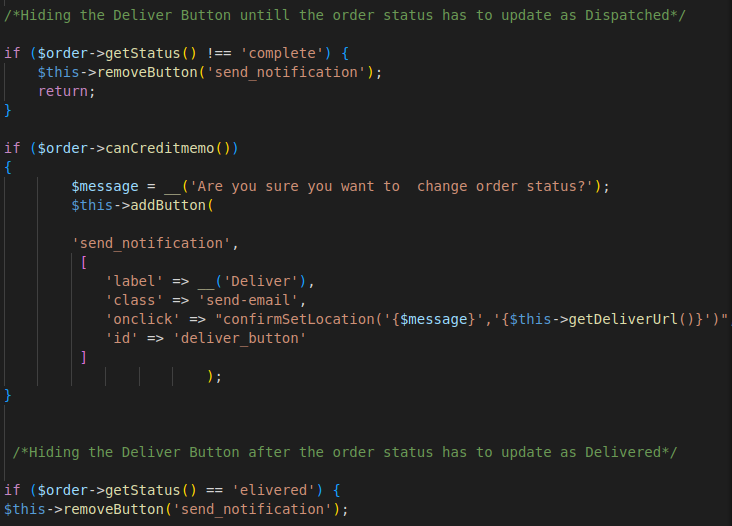
* Initially, to create a new button we have to override core module exists in

Magento/Sales/Block/Adminhtml/Order/View.php

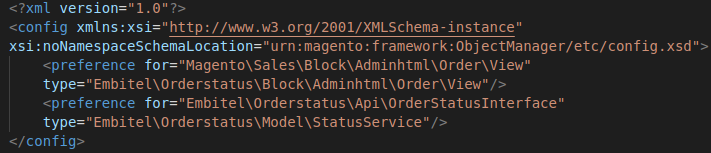
we need to follow the same path to override the View.php files,

The path followed in our module-

**Embitel/Orderstatus/Block/Adminhtml/Order/View.php**

****

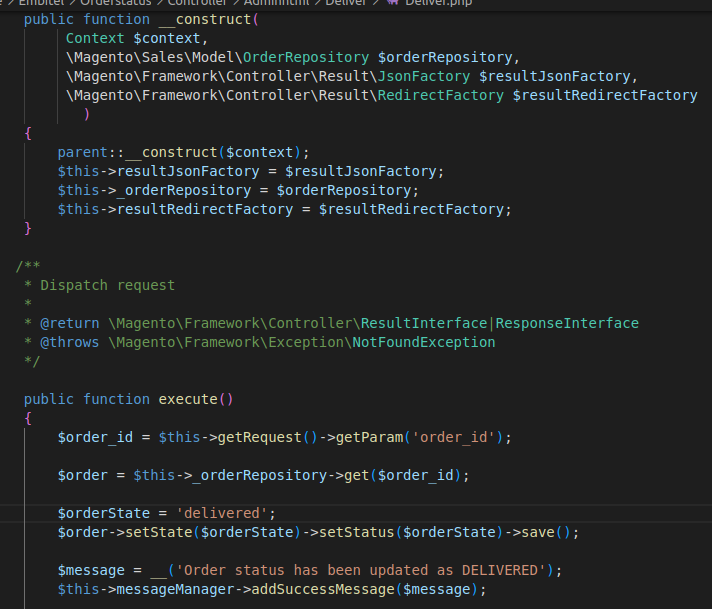
* Next we created di.xml file inside etc folder .This file is used to declare a preference for your new class. A preference is a way of telling Magento 2 to use your new class instead of the original class.



* Then we have created Controller folder to change order status and redirect to the sales-order-view page while clicking on new button named “Deliver”.

The path of the file is-

**Embitel/Orderstatus/Controller/Adminhtml/Deliver/Deliver.php**



* **Creating REST Api and GraphQl to update the order status as “DELIVERED”.**

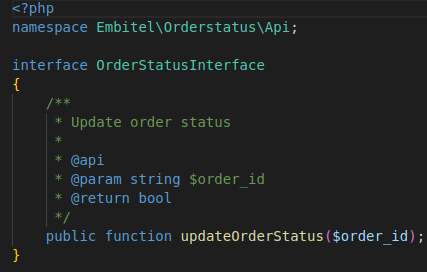
**1. REST Api :**

* In this project we have created the Api request that we send via url,

First we have to create interface file where method signature will be exists,

The path of the file is-

**Embitel/Orderstatus/Api/OrderStatusInterface.php**

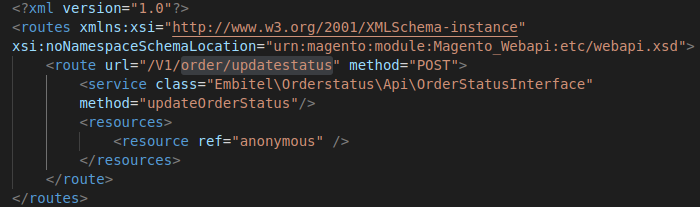
****

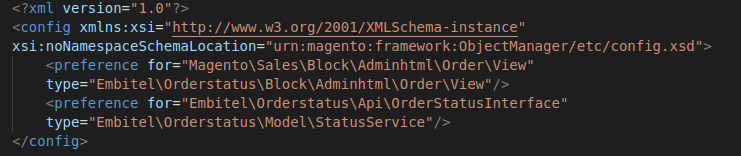
****

* After creation of interface then we have to create the webapi.xml under etc folder. In these file we have url and method of request.
* The di.xml file is typically placed in the etc folder of a module, and it is used to configure the dependency injection container.

etc/webapi.xml

etc/di.xml

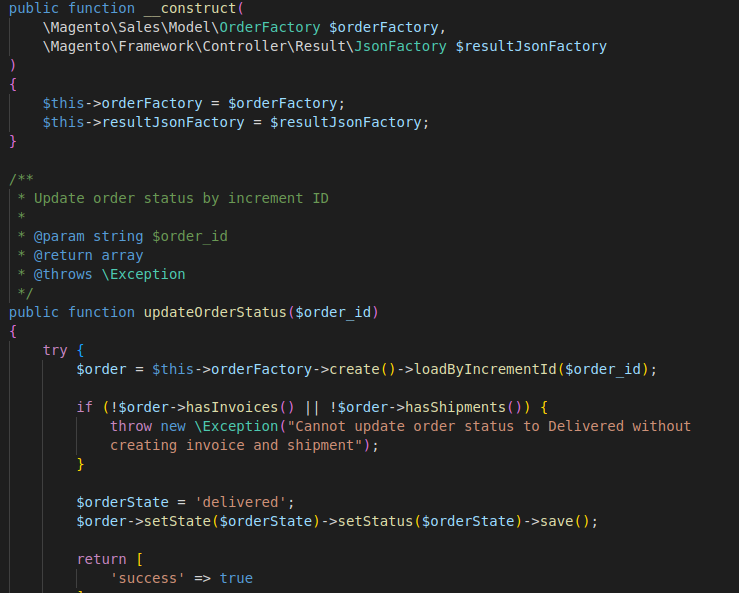




* Then we have created StatusService.php to handle the API request and perform the necessary logic to update the order status.

The path is-

**Embitel/Orderstatus/Model/StatusService.php**



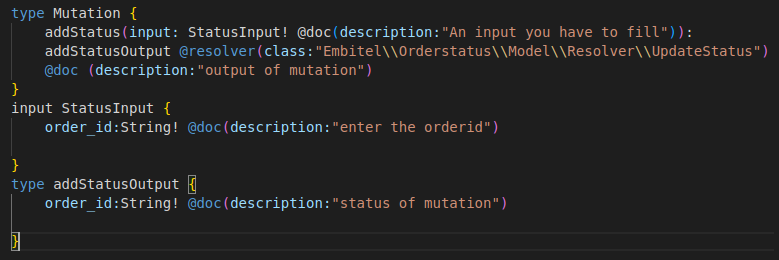
**2. GraphQl :**



* First we have created schema.graphqls under the etc folder.

The path of the file is-

**Embitel/Orderstatus/etc/schema.graphql**



* Next step is, we created the resolver folder inside Model and defined UpdateStatus class inside UpdateStatus.php file.

The path of the file is-

**Model/Resolver/UpdateStatus.php**

