Build DNNrocket Module

# Introduction

This will describe how to create an administration interface for DNNrocket.

# Start

Create Visual Studio project in the /DesktopModules/DNNrocketModules. With a “.Net Standard” class library.

Rename class library to “startconnect.cs”, just to give a standard name across modules.

Add existing projects “DNNrocket API” and “Simplisity” (under ..\DNNrocket) and set a reference.

The startconnect class should inherit from “DNNrocketAPI.APIinterface”.

The DNNrocketAPI project has example “TestForm” and “TestList” to get you started. But other, real life projects would be better examples.

# Themes

Create a “Themes>w3-config>default” folder. This is where the razor template will be kept. Subportals of “confgi-w3\css” and “confgi-w3\js” can be created if required.

# Razor Templates

Create a razor template under the “Themes>w3-config>default”, this will be your default template. The name can be anything you want. Usually there will be multiple razor templates per interface.

# Admin.html or Admin.Aspx

If you don’t; want to create a DNN module you can create a “Admin.html” file. This will be the admin start point of the modules and will be defined in the DNNrocket systemadmin record as “Admin URL”.

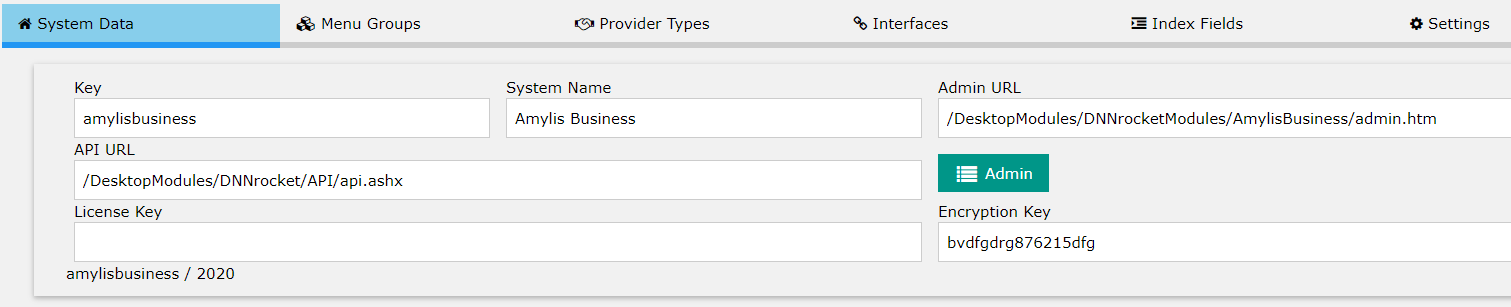
If you need DNN module interaction you need to create a Admin.aspx. this is so we can get the url parameters and pass them to the API code. When creating a DNN module using the DNNrocket API we need to have 1 DNN module for each interface that will be required to interact as a DNN module. The “<moduleName>” must match the system Interface and the “<friendlyName>” of the definition must match the systemprovider name. This is how the DNN modules are linked to the DNNrocket API system. (See RocketMod for an example)

\*\*\*WARNING\*\*\* DNN security is linked to the admin by using the TabId and ModuleId. (Pl,us the DNNrocket roles). Do NOT forget to do this DNN security link for modules.

# Create System

To create the system use “/Desktopmodules/dnnrocket/adminsystem.html”.

Add new system. (Example Below)



**Key**: Unique key for system.

**System Name**: Display name

**Admin URL**: start point of admin interface.

**API url**: API interface url. Usually this will be the rocket API, which will then redirect to the correct system. “/DesktopModules/DNNrocket/API/api.ashx”.

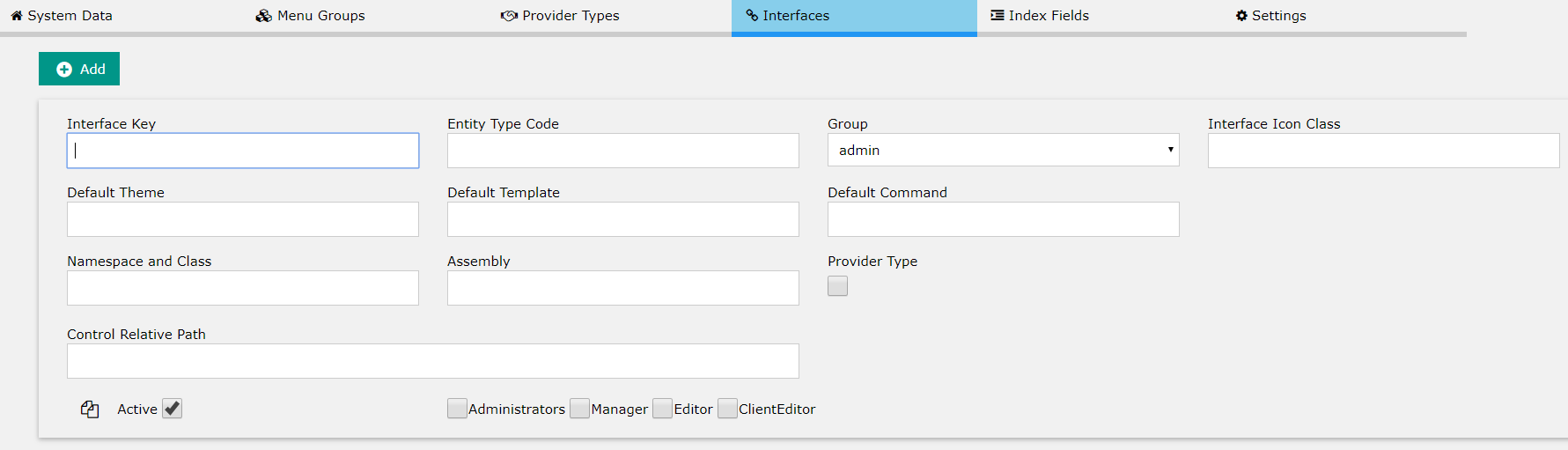
**Licence Key**: License.

**Encryption Key**: This is used to encrypt data in the system and is required.

# Menu Groups

Menu groups appear on the DNNrocket side menu in the admin panel. The side menu is a 2nd level structure, usually the menu groups are the first level, the interfaces are the second level. However, if you do not select a group on the interface, the interface becomes a 1st level link.

# Admin Interface



**Interface Key**: Unique key for the interface.

**Entity Type Code**: The data entity TypeCode, which is the database TypeCode that simplicity uses.

**Group**: Menu Group.

**Interface Icon Class**: Icon.

**Default Theme**: Theme folder which is used by default. The is a single folder under the “Themes” folder of the “Control Relative Path”.

**Default Template**: Default Template. Usually the starting template.

**Default Command**: Default Command. Usually the starting command.

**Namespace and Class**: The namespace and class that the provider will use. This is the namespace and class of the DNNrocket provider.

**Assembly**: The assembly name of the DNNrocket provider. This is used to initiate the assembly.

**Provider Type**: Selected provider type, system defined.

**Control Relative Path**: The relative path to the “Themes” folder and the interface root folder.

**Security**: The default security roles for the interface.