Instalar el Bundle FosJSRouting

Este bundle hace visibles o mejor dicho traduce las rutas del servidor al cliente y pueden ser llamadas desde JavaScript tal cual como se haría desde el servidor.

1. **Ejecutar en línea de comando en el directorio de la app**

```
composer require friendsofsymfony/jsrouting-bundle
```

2. **Habilitar el bundle en "app/AppKernel.php"**

```
$bundles = array(
      ...
      new FOS\JsRoutingBundle\FOSJsRoutingBundle()
```

3. **Registrar la ruta del nuevo bundle en "app/config/routing.yml"**

```
fos_js_routing:
    resource: "@FOSJsRoutingBundle/Resources/config/routing/routing.xml"
```

4. **Publicar los "assets" en linea de comando en el directorio de la app**

```
# Symfony 2
php app/console assets:install --symlink web

# Symfony 3
php bin/console assets:install --symlink web
```

DHTMLX Windows

Librería JavaScript para desplegar ventanas botones y controles muy amigables.

1. **Descargar el componente "Modal" (para hacer las ventanas)**

   ```
   https://dhtmlx.com/x/download/regular/dhtmlxWindows.zip
   ```

2. **Dentro hay un directorio "codebase", de él sacar los archivos css y js y ponerlos dentro de "public/js" y "public/css" respectivamente**

   ```
   public/css/dhtmlxwindows.css
   public/js/dhtmlxwindows.js
   ```

3. **Los directorios "fonts" y "imgs" copiarlos tal cual a "public/css"**

   ```
   public/css/fonts
   public/css/imgs
   ```

4. **Agregar el js y css como assets en el layout principal "app/Resources/views/layout.html.twig"**

   ```
   # Después del asset "BootStrap"
   <link rel="stylesheet" href="{{ asset('public/css/dhtmlxwindows.css') }}" />

   # Después del asset jQuery
   <script type="text/javascript" src="{{ asset('public/js/dhtmlxwindows.js') }}"></script>
   ```

Integración

FosJSRouting y DHTMLX Windows trabajarían en conjunto para lograr hacer aparecer una ventana completamente funcional e independiente sobre la vista actual.

1. **Convertir la ruta en "ruta a exponer", o sea convertirla para que pueda ser vista dentro de una ventana, para este caso, emplearemos como ejemplo mostrar en ventana la "edición" de un registro de menú, por ello convertiremos en ruta a exponer la edición**

```
# src/MRF/MenuBundle/Resources/config/routing.yml
mrf_menu_edit:
    path:     /Menu/edit/{id}
    defaults: { _controller: MRFMenuBundle:Menu:edit }
    options:
        expose: true
```

2. **Modificar el actual llamado a la vista de edición para que aparezca ahora dentro de una ventana "src/MRF/MenuBundle/Resources/views/Menu/list.html.twig"**

```
# Antes
<a class="glyphicon glyphicon-pencil" href="{{ path('mrf_menu_edit', {id: Menu.id}) }}"></a>

# Ahora
<a class="glyphicon glyphicon-pencil" onclick="editarMenu({{ Menu.id }})"></a>
```

3. **Como se puede ver, ahora llamamos a una función "editarMenu" y le brindamos como parámetro el ID del menú, creamos esta función dentro de etiquetas "<script>" dentro del bloque "{% block seccion3 %}"**

```
# src/MRF/MenuBundle/Resources/views/Menu/list.html.twig
function editarMenu(id) {

  wins = new dhtmlXWindows();
  skin = "dhx_terrace";
  wins.setSkin(skin);

  winId = "winMenuEditar";

  wins.createWindow(winId, 1, 1, 780, 530);
  wins.window(winId).setText("Modificar Menu");

  wins.window(winId).setModal(true);
  wins.window(winId).button("minmax").disable();
  wins.window(winId).centerOnScreen();
  wins.window(winId).progressOn();

  wins.window(winId).setIconCss("without_icon");

  var url = Routing.generate('mrf_menu_edit', {'id': id});
  wins.window(winId).attachURL(url);

  wins.window(winId).attachEvent("onContentLoaded", function(id){
     id.progressOff();
  });

}
```

4. **Como se puede notar será necesario hace un layout solo para ventanas modales**

```
# app/Resources/views/layoutWindow.html.twig
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <meta   name="viewport"   content="width=device-width,   user-scalable=no,   initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
        <title>{% block title %}Mantenedores!{% endblock %}</title>
        {% block stylesheets %}
            <link rel="stylesheet" href="{{ asset('public/css/bootstrap.min.css') }}" />
            <link rel="stylesheet" href="{{ asset('public/css/style.css') }}"/>
        {% endblock %}
    </head>
    <body>
        <section class="main container" id="menus">
            {% block seccion1 %}
            {% endblock %}
        </section>
    </body>
</html>
```

5. **La vista de edición tiene que ser recortada para que solo muestre la edición, o sea sin el menú, ni banner, ni usuario conectado y la insertaremos en el nuevo layout**

```
# src/MRF/MenuBundle/Resources/views/Menu/edit.html.twig
{% extends 'layoutWindow.html.twig' %}
{% block seccion1 %}
    {{ form_start(form)}}
        <div >
        {{ form_label(form.nombre) }}
        {{  form_widget(form.nombre, {'attr' : {'placeholder' : 'Nombre', 'class': 'form-
control'}}) }}
        <span >{{ form_errors(form.nombre) }}</span>
        </div>

        <div >
        {{ form_label(form.ruta) }}
        {{ form_widget(form.ruta, {'attr' : {'placeholder' : 'Ruta', 'class': 'form-control'}})
}}
        <span >{{ form_errors(form.ruta) }}</span>
        </div>
        <div >
        {{ form_label(form.listar) }}
        {{ form_widget(form.listar, { 'attr': {'checked': 'checked', 'class': 'form-control'}
}) }}
        <span >{{ form_errors(form.listar) }}</span>
        </div>
        <div >
        {{ form_label(form.agregar) }}
        {{  form_widget(form.agregar, {'attr' : {'placeholder' : 'Agregar', 'class': 'form-
control'}}) }}
        <span >{{ form_errors(form.agregar) }}</span>
        </div>
        <div >
        {{ form_label(form.editar) }}
        {{  form_widget(form.editar, {'attr' : {'placeholder' : 'Editar', 'class': 'form-
control'}}) }}
        <span >{{ form_errors(form.editar) }}</span>
        </div>

        <div >
        {{ form_label(form.eliminar) }}
        {{ form_widget(form.eliminar, {'attr' : {'placeholder' : 'Eliminar', 'class': 'form-
control'}}) }}
        <span >{{ form_errors(form.eliminar) }}</span>
        </div>
        <br>
```

```
            <p>
                {{ form_widget(form.save, {'label' : 'Actualizar Menu', 'attr' : { 'class': 'form-
control', 'class': 'btn btn-success'}}) }}
            </p>
        {{form_end(form)}}
    {% endblock %}
```

6. **Finalmente, al presionar "Actualizar Menu" en la ventana creada, se mostrará la vista listado sobre esta, debemos cambiar esto en el controlador "src/MRG/MenuBundle/Controller/MenuController.php"**

```
# Antes
public function UpdateAction($id, Request $request){
      ...
      if($form->isSubmitted() && $form->isValid()){
      $em->flush();
      $this->addFlash('mensaje', 'El Menu fue Actualizado con exito.');
      return $this->redirectToRoute('mrf_menus_list', array('id'=>$menu->getId()));
      }
      ...
}

# Ahora
public function UpdateAction($id, Request $request){
      ...
      if($form->isSubmitted() && $form->isValid()){
      $em->flush();
      $this->addFlash('mensaje', 'El Menu fue Actualizado con exito.');
      return $this->redirectToRoute('mrf_menu_edit', array('id'=>$menu->getId()));
      }
      ...
}
```