Plugin Development: Routes

Routes allow you to create additional pages in the system that can only be accessed when the plugin is **active**. All of your routes must be placed in one or several PHP files in the Routes folder. You can register a route for any HTTP verb, the most common Router methods are get(), post(), put() and delete().

The following filenames have special meaning within the Routes folder and will inherit specific attributes:

Filename	Attributes
frontend.php	Authenticated routes for the frontend. Routes de mode is enabled.
unauthenticated.php	Unauthenticated routes for the frontend. Routes mode is enabled.
operator.php	Authenticated operator panel routes. Routes defi name.
settings.php / plugin.php / widget.php	Authenticated operator panel routes. Routed defi above Settings > Admin Folder) and also be affe
api.php	Authenticated API routes. Routes defined in this
nolocale.php	The same functionality as exc

POST, PUT and DELETE routes in any of the above predefined route files require the a CSRF token for the request to be processed successfully. This can be added to forms with the csrf_field or csrf_token helper functions.

```
<form method="POST" action="/route">    {{ csrf_field() }}    <!--
Outputs: <input name="_token" type="hidden" value=
```

```
"rzV9ynHksMChagph6DpQJTY3tLQaRSyggQeM1Z9e"> -->    ...</form>
```

```
<form method="POST" action="/route">    <input type="hidden"
name="_token" value="{{ csrf_token() }}" />    ...</form>
```

Other Route Files

Files with names other than those listed in the above table do not inherit any attributes. You're free to create whatever routes you require, for example unauthenticated routes.

Defining Routes

A very simple route looks like this.

```
$router->get('settings', [    'as' =>
'plugin.helloworld.settings',    function () {        return 'Hello World';    }]);
```

The first parameter of the method is the URI, and this is relative to the plugin URL in the operator panel. For example if your help desk is hosted at http://domain.com/support/ then the above route would correspond
to http://domain.com/support/admin/plugin/helloworld/settings.

The second parameter is an array of attributes for the route, the 'as' key allows you to name the route which makes it easier to use in controllers and views. The name needs to be unique to not conflict with any other routes, and we recommend to start your routes with 'plugin.pluginname' to ensure this doesn't happen.

A closure was used in the above example for simplicity, allowing to define what should be shown on the route in this file, but we recommend to instead use the 'uses' key to define a controller action that is called when the route is matched. A controller action is simply a function in a controller file and this promotes keeping the code clean by ensuring all logic is placed in the controllers. Below are the example routes included in the skeleton plugin that display the settings page and handle updating the settings, these call the getSettingsPage and updateSettings functions in the main controller.

Routes/plugin.php

```
<?php$router->get('settings', [    'as'   =>
 'plugin.helloworld.settings',    'uses' =>
 'Addons
PluginsHelloWorl
dControllersHelloWorld@getSetting
sPage']);$router->post('settings', [    'as'   =>
 'plugin.helloworld.settings.update',    'uses' =>
 'AddonsPluginsHelloWorldControllersHelloWorld@updateSettings']);
```

More options and documentation on routing is available at the [redacted].

Language Detection

If you make use of [redacted], language detection is in place which will attempt to detect the locale and translate available content. For example, if you have Spanish enabled and your custom plugin route is https://domain.com/support/helloworld/myroute then the following options are available:

URL
https://domain.com/**es**/support/helloworld/myroute
https://domain.com/support/helloworld/myroute**?lang=es**
https://domain.com/support/helloworld/myroute

For API routes, you must use the lang parameter to set a language other than the system default, e.g. https://domain.com/support/api/plugin/helloworld/myroute?lang=es.

Routes can be accessed via their defined name under the as attribute described above. For example, to redirect to the plugin settings page, defined above, in a controller you would use:

```
Redirect::route('plugin.helloworld.settings')
```

To link to the plugin Hello World settings page in a view, you would use:

```
{{ route('plugin.helloworld.settings') }}
```

Online URL:
https://docs.supportpro.vn/article/plugin-development-routes-207.html