

Adding Admin Settings to your Plugin

NB: these are rough notes, and could do with checking and/or tidying up from someone who understands plugins better

If for instance you want to give users access to certain settings/configurations that your plugin might have, there are two ways of achieving this:

1. A Kohana config file

Create a config folder within your plugin folder. Within this folder create a config file with a unique file name example: [my_plugin_name].php. This configuration file must adhere to [Kohana config](#) file standards.

Example: config/youtube.php

```
$config['items_per_page'] = 10;
```

This setting will then be available anywhere in the Ushahidi application using

```
Kohana::config('youtube.items_per_page');
```

2. Ushahidi Admin Console

This method requires you to create a separate database table to store your settings in.

You need to create one table and four new files to add your admin console. Please use the method described in [\[How to Write A Plugin\]](#) to create your table. The other things you need to add are:

- A new table in libraries/myplugin_install.php where [my_plugin] is the unique name of your plugin.
- A new model in models/myplugin_settings.php
- A new controller called [my_plugin]_settings.php.
- A new view in views/myplugin/admin/myplugin_settings.php

Like this:

```
- [plugin folder]
| - [my_plugin]
|   - controllers
|     - admin
|       [my_plugin]_settings.php
|   - libraries
|     - [my_plugin]_install.php
|   - models
|     - [my_plugin]_settings.php
|   - views
|     - [my_plugin]
|       - admin
|         [my_plugin]_settings.php
```

The Ushahidi system will automatically detect this controller and will display a [Settings] link to this controller from the plugin list in the admin console. Once you create all these files, the controller has all the functionality necessary to save and update settings in your my_plugin settings table.

First, create a database table for your settings. You put this into libraries/myplugin_install.php. For example,

```

class myplugin_Install {/**
    Constructor to load the shared database library*/
    public function __construct()
    {
        $this->db = Database::instance();
    }

    /**
    Creates the required database tables for the plugin*/
    public function run_install()
    {
        // Create the database tables.
        // Also include table_prefix in name
        $this->db->query("CREATE TABLE IF NOT EXISTS `".Kohana::config('database.default.table_prefix')."myplugin_settings`
(id int(11) unsigned NOT NULL AUTO_INCREMENT,
    myplugin_title varchar(40) DEFAULT NULL,
    myplugin_text varchar(400) DEFAULT NULL,
    PRIMARY KEY (`id`)
);
");
    }

    /**
    Deletes the database tables for the actionable module*/
    public function uninstall()
    {$this->db->query("DROP TABLE `".Kohana::config('database.default.table_prefix')."myplugin_settings`");
    }
}

```

Then create a model that matches that database table. You put this into models/myplugin_settings.php

```

class Myplugin_Settings_Model extends ORM
{
    // Database table name
    protected $table_name = 'myplugin_settings';
}

```

Then create an admin controller in file controllers/admin/myplugin_settings.php To use the settings values, add this into your controller:

```

$settings = ORM::factory('myplugin_settings')->find(1);
$view->myplugin_title = $settings->myplugin_title;
$view->myplugin_text = $settings->myplugin_text;

```

Then create an admin view in views/myplugin/admin/myplugin_settings.php. Add this into your view:

```
<br />
<div class="fancy">
<h5><?php echo $myplugin_title; ?></h5>
<?php echo $myplugin_text; ?><br />
</div>
```

Now you can go to the settings page and set these values.