Settings

- Overview
- Function Reference
- Adding Setting Fields
- Adding Settings Sections
- Registering Settings
- Options Form Rendering

Overview

The **Settings API**, added in WordPress 2.7, allows admin pages containing settings forms to be managed semi-automatically. It lets you define settings pages, sections within those pages and fields within the sections.

New settings pages can be registered along with sections and fields inside them. Existing settings pages can also be added to by registering new settings sections or fields inside of them.

Organizing registration and validation of fields still requires some effort from developers using the Settings API, but avoids a lot of complex debugging of underlying options management.

NOTE: When using the Settings API, the form posts to `wp-admin/options.php` which provides fairly strict capabilities checking. Users will need `manage_options` capability (and in MultiSite will have to be a Super Admin) to submit the form.

The functions are found in `wp-admin/includes/plugin.php` and `wp-admin/includes/template.php`

Function Reference

Setting Register/Unregister:

- register_setting()

- unregister_setting()

Add Field/Section:

- add_settings_field()
- add_settings_section()

Options Form Rendering:

- settings_fields()
- do_settings_sections()
- do_settings_fields()

Errors:

- add_settings_error()
- get_settings_errors()
- settings_errors()

Adding Setting Fields

You can add new settings fields (basically, an option in the `wp_options` database table but totally managed for you) to the existing WordPress pages using this function. Your callback function just needs to output the appropriate HTML input and fill it with the old value, the saving will be done behind the scenes. You can create your own sections on existing pages using `add_settings_section()` as described below.

NOTE: You MUST register any options you use with `add_settings_field()` or they won't be saved and updated automatically. See below for details and an example.

```
add_settings_field( $id, $title, $callback, $page, $section = 'default', $args = array() )
```

- `$id` – String for use in the 'id' attribute of tags.
- `$title` – Title of the field.
- `$callback` – Function that fills the field with the desired inputs as part of the larger form. Name and id of the input should match the $id given to this function. The function should echo its output.
- `$page` – The type of settings page on which to show the field (general, reading, writing, ...).

- `$section` – The section of the settings page in which to show the box (default or a section you added with add_settings_section, look at the page in the source to see what the existing ones are.)
- `$args` – Extra arguments passed into the callback function

Adding Settings Sections

Settings Sections are the groups of settings you see on WordPress settings pages with a shared heading. In your plugin you can add new sections to existing settings pages rather than creating a whole new page. This makes your plugin simpler to maintain and creates fewer new pages for users to learn. You just tell them to change your setting on the relevant existing page.

```
add_settings_section( $id, $title, $callback, $page );
```

- `$id` – String for use in the 'id' attribute of tags.
- `$title` – Title of the section.
- `$callback` – Function that fills the section with the desired content. The function should echo its output.
- `$page` – The type of settings page on which to show the section (general, reading, writing, media etc.)

Registering Settings

```
register_setting( $option_group, $option_name, $args );
```

```
unregister_setting( $option_group, $option_name );
```

NOTE: `register_setting()` as well as the above mentioned `add_settings_*()` functions should all be called from a `admin_init` action hook callback function. Refer to the "Examples" section below.

Options Form Rendering

When using the API to add settings to existing options pages, you do not need to be concerned about the form itself, as it has already been defined for the page. When you define a new page from scratch, you need to output a minimal form structure that contains a few tags that in turn output the actual sections and settings for the page.

To display the hidden fields and handle security of your options form, the Settings API provides the settings_fields() function. `settings_fields( $option_group );`

`$option_group` **(*string*) (*required*):**

A settings group name. This must match the group name used in register_setting(), which is the page slug name on which the form is to appear. Default: *None*

To display the sections assigned to the page and the settings contained within, the Settings API provides the do_settings_sections() function. `do_settings_sections( $page );`

`$page` **(*string*) (*required*):**

The slug name of the page whose settings sections you want to output. This should match the page name used in add_settings_section(). Default: *None*

The do_settings_fields() function is provided to output the fields assigned to a particular page and section. You should not call this function directly, rather use `do_settings_sections()` to output the Section content as well as the associated fields.

Your options form also needs a submit button. You can use the submit_button() function to do this.

Finally, you need to output the HTML

□□□ #1
□ □□□ □□□ 2026-08-29 23:36:33 UTC
□ □□□ □□□ 2026-08-29 23:36:33 UTC