

Posting to Drupal

This lesson will show you how to post to Drupal using ScreenSteps. This lesson uses Drupal version 6.2.

Navigate to Drupal Administration Area



The screenshot shows the Drupal administration interface for Blue Mango Learning Systems. The user 'trevordevore' is logged in. The 'Administer' menu item is highlighted with a red box. The main content area shows a warning message and a 'Content management' section.

Home

Administer **By task** By module

One or more problems were detected with your D

Welcome to the administration section. Here you r

Hide descriptions

Content management
Manage your site's content.

Comments
List and edit site comments and the comment n
queue.

Content

Enable Blog API

ing
he

Themes

Change which theme your site uses or allows users to set.

needs

Site configuration

Adjust basic site configuration options.

our

Actions

Manage the actions defined for your site.

Administration theme

Settings for how your administrative pages should look.

res.

Blog API

Configure the content types available to external blogging clients.

Clean URLs

Enable or disable clean URLs for your site.

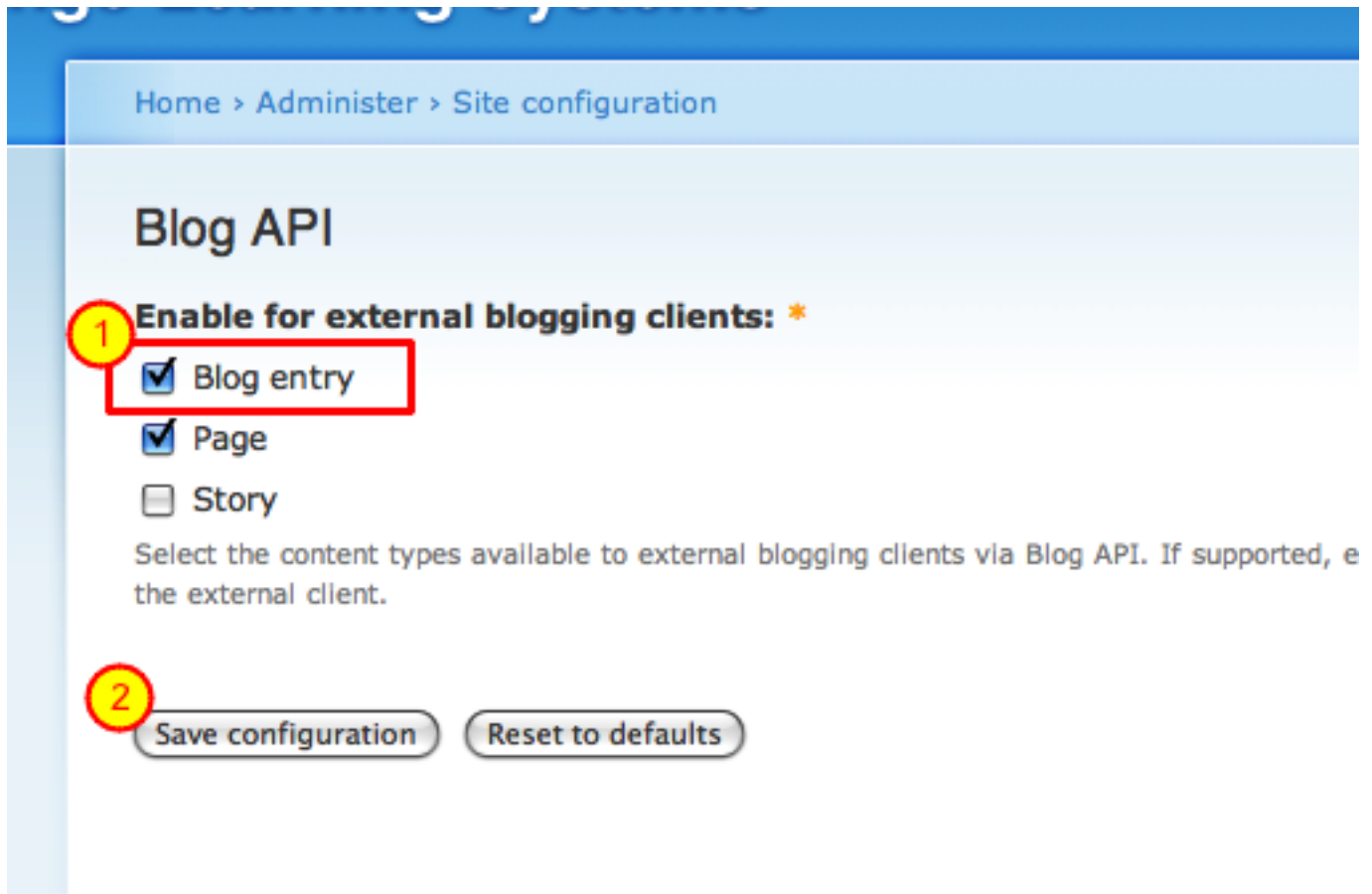
Date and time

Settings for how Drupal displays date and time, as well as the system's default timezone.

for

Navigate to Site Configuration->Blog API.

Enable Blog Entry



Home > Administer > Site configuration

Blog API

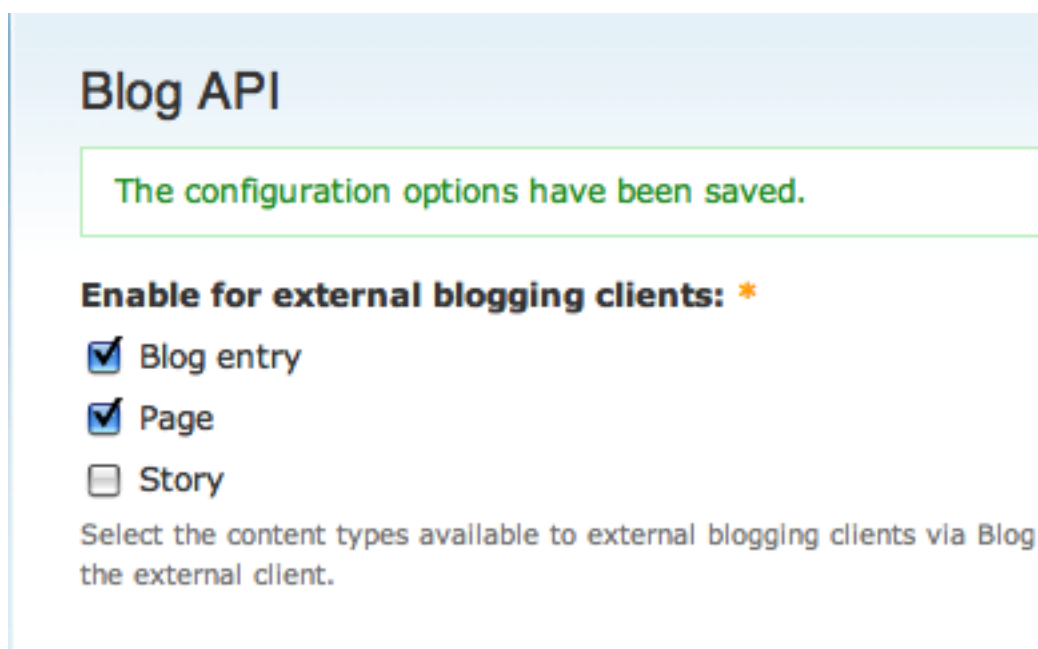
1 Enable for external blogging clients: *

- Blog entry
- Page
- Story

Select the content types available to external blogging clients via Blog API. If supported, e the external client.

2 Save configuration Reset to defaults

Make sure the **Blog Entry** checkbox is checked (1) and click **Save Configuration** (2).



Blog API

The configuration options have been saved.

Enable for external blogging clients: *

- Blog entry
- Page
- Story

Select the content types available to external blogging clients via Blog the external client.

Your settings will be saved.

Update blogapi_metaweblog_get_category_list

```

/**
 * Blogging API callback. Returns a list of the taxonomy terms that can be
 * associated with a blog node.
 */
function blogapi_metaweblog_get_category_list($blogid, $username, $password) {
  // BEGIN FIX: Without validating the user, the _blogapi_validate_blogid will
  // return an empty array
  $user = blogapi_validate_user($username, $password);

  if (!$user->uid) {
    return blogapi_error($user);
  }
  // END FIX

  if (($error = _blogapi_validate_blogid($blogid)) !== TRUE) {
    // Return an error if not configured type.
    return $error;
  }

  $vocabularies = module_invoke('taxonomy', 'get_vocabularies', $blogid, 'vid');
  $categories = array();
  if ($vocabularies) {
    foreach ($vocabularies as $vocabulary) {
      $terms = module_invoke('taxonomy', 'get_tree', $vocabulary->vid, 0, -1);
      foreach ($terms as $term) {
        $term_name = $term->name;
        foreach (module_invoke('taxonomy', 'get_parents', $term->tid, 'tid') as $parent) {
          $term_name = $parent->name . '/' . $term_name;
        }
        $categories[] = array('categoryName' => $term_name, 'categoryId' => $term->tid);
      }
    }
  }
  return $categories;
}

```

In Drupal version 6.2 there appears to be some missing code in the `blogapi_metaweblog_get_category_list` function. This function is called when ScreenSteps attempts to retrieve available categories for a blog. Drupal reports a server error when it shouldn't. The fix is easy and here is how to do it:

- 1) Open **blogapi.module** which is located in the `./modules/blogapi/` folder of your Drupal installation.
- 2) Locate the `blogapi_metaweblog_get_categorylist` function. In my installation it starts on line 392.
- 2) Add the following code in the spot specified in the image:

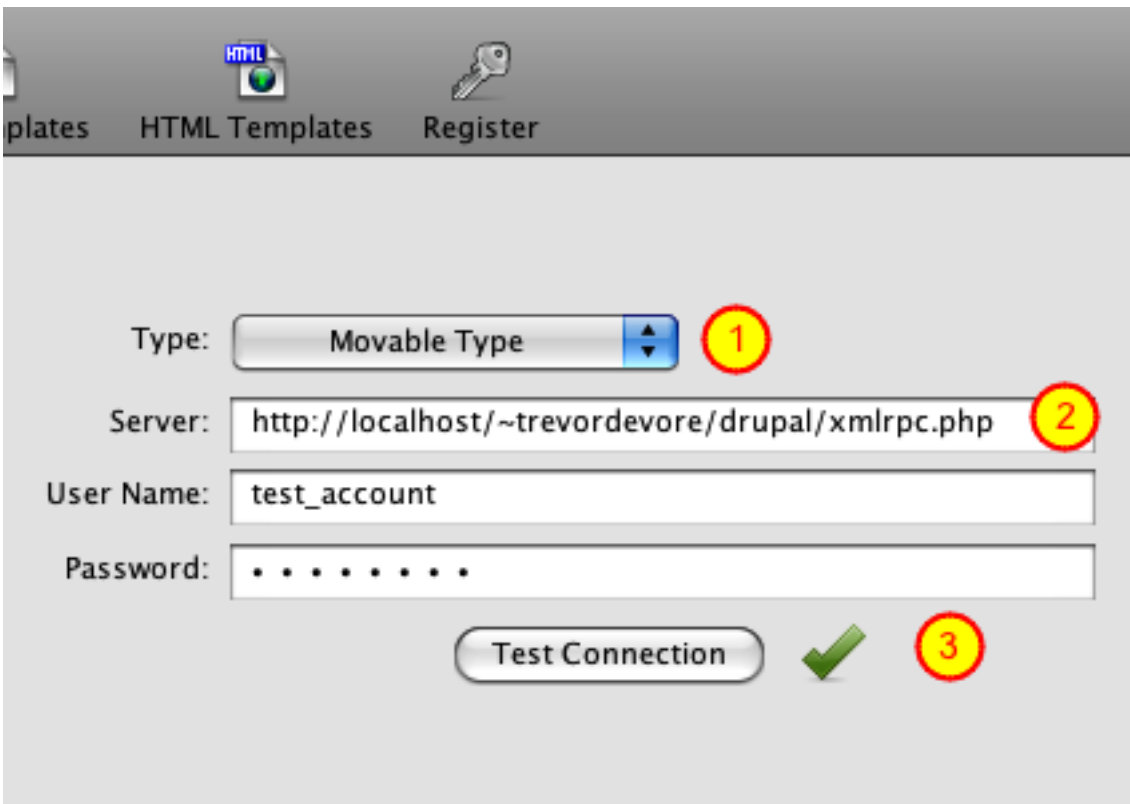
```

// BEGIN FIX: Without validating the user, the _blogapi_validate_blogid will
// return an empty array

```

```
$user = blogapi_validate_user($username, $password);  
  
if (!$user->uid) {  
    return blogapi_error($user);  
}  
// END FIX
```

Configure ScreenSteps



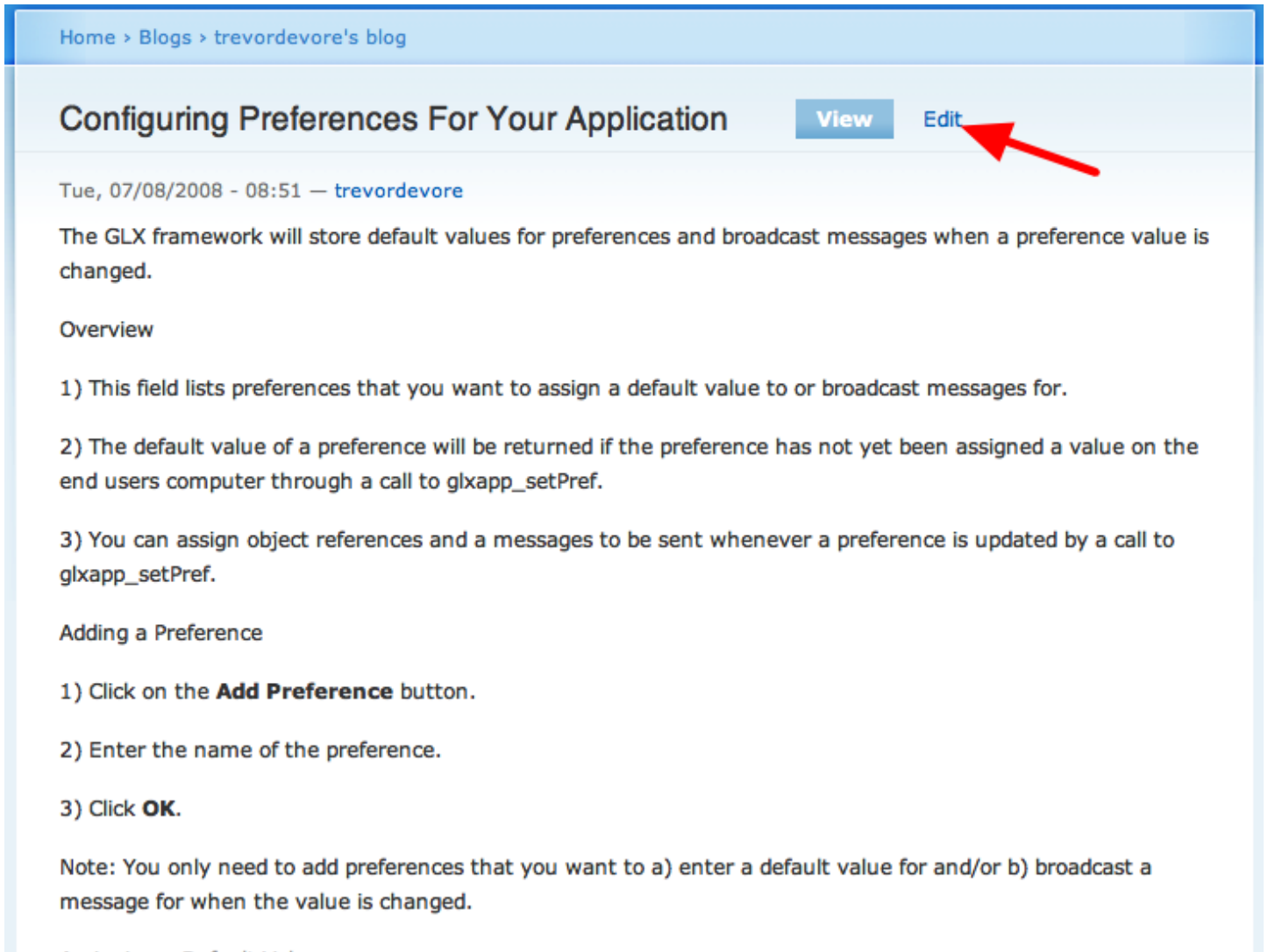
The screenshot shows a configuration window with a grey header containing icons for 'HTML Templates' and 'Register'. Below the header are four input fields: 'Type' (a dropdown menu set to 'Movable Type'), 'Server' (a text field containing 'http://localhost/~trevordevore/drupal/xmlrpc.php'), 'User Name' (a text field containing 'test_account'), and 'Password' (a masked text field with dots). Below these fields is a 'Test Connection' button, a green checkmark icon, and a red circle with the number '3'. Red circles with numbers '1' and '2' are also present, highlighting the 'Type' dropdown and the 'Server' text field respectively.

In order to post to Drupal you should configure a Web Account ([instructions here](#)) using the Movable Type API (1). In the root folder of your Drupal installation is a file named **xmlrpc.php**. The full url to the file should be entered in the **Server** field (2).

Enter your user name and password and click **Test Connection** (3) to verify that everything is configured correctly.

You can now post your lesson to Drupal following [these instructions](#).

Watch Out For Input Formatters



Home > Blogs > trevordevore's blog

Configuring Preferences For Your Application [View](#) [Edit](#)

Tue, 07/08/2008 - 08:51 — trevordevore

The GLX framework will store default values for preferences and broadcast messages when a preference value is changed.

Overview

- 1) This field lists preferences that you want to assign a default value to or broadcast messages for.
- 2) The default value of a preference will be returned if the preference has not yet been assigned a value on the end users computer through a call to `glxapp_setPref`.
- 3) You can assign object references and a messages to be sent whenever a preference is updated by a call to `glxapp_setPref`.

Adding a Preference

- 1) Click on the **Add Preference** button.
- 2) Enter the name of the preference.
- 3) Click **OK**.

Note: You only need to add preferences that you want to a) enter a default value for and/or b) broadcast a message for when the value is changed.

Assigning a Default Value

After posting my first lesson to drupal I opened the blog post in the browser but no images appeared. This is because the **Filtered HTML** input formatter was the default for all content. To fix this click on the **Edit** link.

Change Input Format to Full HTML

<p>2) The default value of a preference will be returned if the preference has not yet value on the end users computer through a call to glxapp_setPref. </p>

Input format 1

Filtered HTML

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <cite> <code> <dl> <dt> <dd>
- Lines and paragraphs break automatically.

Full HTML 2

- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

[More information about formatting options](#)

▶ [Revision information](#)

▶ [Comment settings](#)

▶ [Authoring information](#)

▶ [Publishing options](#)

3

Click on Input format (1) to expand the available options. Click on the **Full HTML** (2) radio button and click **Save** (3).

Success!

Home > Blogs > trevordevore's blog

Configuring Preferences For Your Application

View

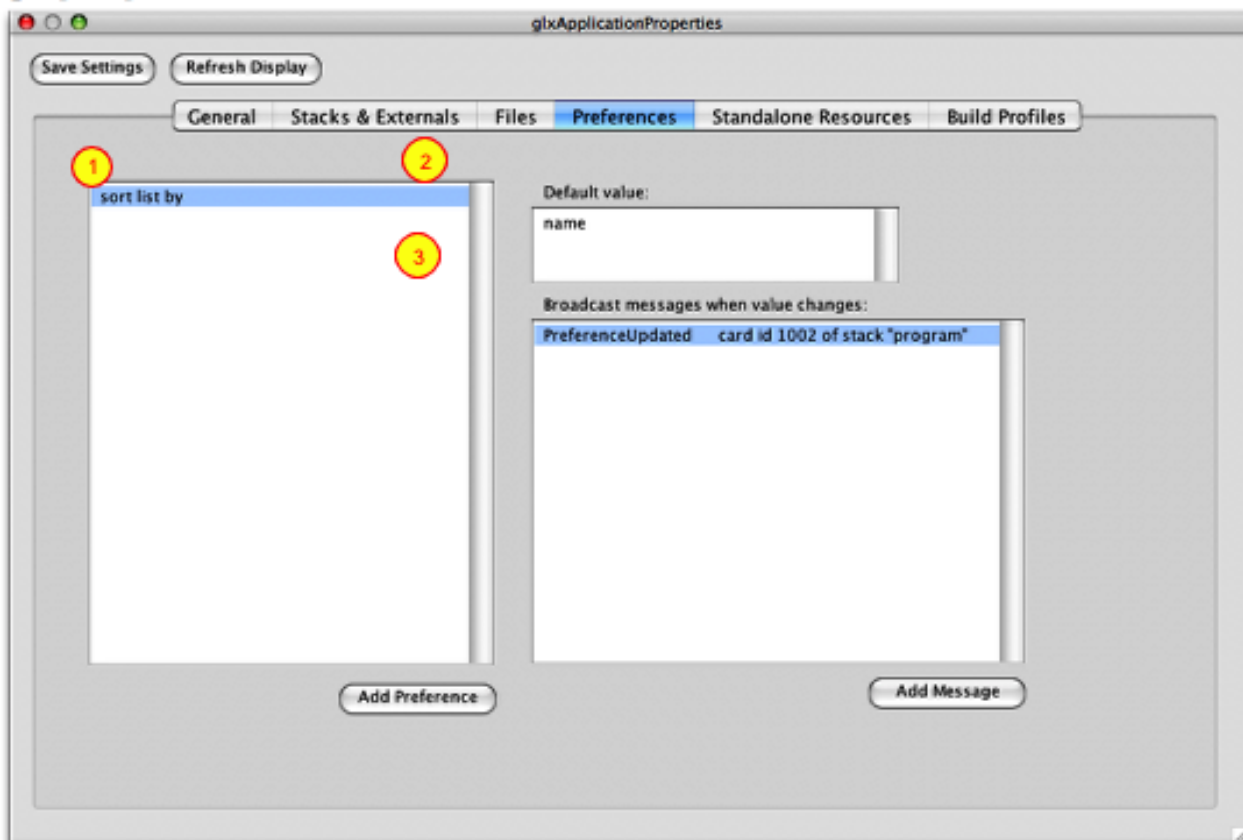
Edit

Blog entry *Configuring Preferences For Your Application* has been updated.

Tue, 07/08/2008 - 08:51 — [trevordevore](#)

The GLX framework will store default values for preferences and broadcast messages when a preference is changed.

Overview



Your lesson now displays correctly in Drupal.