TutorialGraffittiWall

Now as a tutorial let s imagine that we want to write a new feeature to Tiki called graffitti wall , the graffitti wall will display the last n graffitis (configurable) and will provide users with a simple text area were a graffitti can be entered. HTML code will not be allowed.

Planning our module

We ll need one screen for the graffitti wall:

- tiki-graffitti : Display the last n graffittis and provide a form to enter a new graffiti.

The screen will be compunded by:
tiki-graffitti.php : PHP code
tiki-graffitti.tpl : Template

We'll also add a new library to Tiki called graffittilib.php in the lib directory. Since we need to store graffittis somewhere we have to create a table to store graffittis, we ll write the .sql creation script in a file called graffitti.sql this will be our table:

```
drop table if exists tiki_graffittis;
create table tiki_graffittis (
graffittiId integer(12) not null auto_increment,
data text,
timestamp integer(14),
primary key(graffittiId)
);
```

Note: Timestamps and dates are stored in tiki tables as Unix timestamps and are represented as 14 digit long integers. Please use this format to keep the tables consistent.

Execute the .sql script using for example

```
mysql tiki < graffitti.sql
```

The library Now we ll write the library that will be used to access the graffittis table, we ll call the library graffittilib.php and it will be a class where we ll encapsulate all the functions that will be used to display graffittis.

This is the library: ***attach graffittilib.php here***

The form in which this class was written should be followed by all the libraries used in Tiki.

The class constructor receives a pear connection object used to access the database through PEAR in the class, we added a method to display SQL errors and two application methods: add_graffitti($data) that will be used to add a new graffitti and get_graffittis($n) that will return an array with the last n graffittis.

Writing skeletons and putting all together

Now put graffittilib.php in the lib directory.

Create tiki-graffitti.php in the main tiki directory and write the following skeleton:

attach tiki-graffitti.php here

Let s examine the code, tiki-setup is a php file that MUST be included in all the Tiki scripts, in that file permissions and preferences are loaded the database connection is stablished, the session is checked to determine if the user is logged etc. Don t forget to include it in your Tiki modules. Then we included our library from the lib directory. After including the lib we create a GraffitttiLib object passing the global variabl $dbTiki that contains the default database connection as indicated in db/tikidb. php.

Finally we set-up the template to use inthe middle area of Tiki and we display the template tiki.tpl.

Create the skeleton layout tiki-graffitti.tpl and put it in the templates directory.

attach tiki-graffitti.tpl here

Just a title since the templates are normal HTML code mixed with Smarty directives.

Now check if everything is ready to be filled in by accessing tiki-graffiti.php from a browser:

insert graphic

Wow! The tile is displayed and we are ready to code our Graffitti module. If you are an experienced PHP developer and know Smarty this is all you need to start coding new features for Tiki, if you need help or have questions you are free to ask, subscribe to tiki-devel the Tiki developers mailing list and start coding. (You can subscribe to the mailing list from the sourceforge page: http://tikiwiki.sourceforge.net)

If you want to learn more and learn how to code the Graffitti module just keep reading!

Coding the graffitti module

Now, we have the skeleton ready to be filled in, we ll start by adding a form to the template to enter new graffittis.

This is tiki-graffitti.tpl with the form.

new version of tiki-graffitti.tpl

Very important notes:

- Forms usually are processed by the same script that displayed the template containing the form, in this example tiki-graffitti.php
- XHTML should be used note how the *input* tag ends with />
- The `{tr}{/tr}` smarty block must be used to escape strings that can be translated to different languages.

And this is displayed if you access tiki-graffitti.php

insert graphic

Now we are ready to process the addition of a new graffitti to the wall, let s go to tiki-graffitti.php and add the code to process an addition.

new version of tiki-graffitti.php

Note how $_REQUEST is used to access submitted variables, this is to make sure that the script will work in PHP installations where register_globals is turned off. It s very important to respect this in your Tiki modules.

The code is easy if we are processing a submit button named send then call the add_graffitti method in our library to add the graffitti.

So now you can add graffittis but you can t see them!, let s add some code to get the list of graffittis from the database. Add this before the code that processes the form in tiki-graffitti.php

```
$graffittis = $glib->get_graffittis(10);
$smarty->assign_by_ref('graffittis',$graffittis);
```

This gets the last 10 graffittis from the database and assigns them to a smarty template variable called graffittis now we have to display the graffittis in the template. Edit tiki-graffitti.tpl

new version of tiki-graffitti.tpl

Note how the Smarty {section} directive is used to loop the $graffittis template variable assigned in the PHP file, for each graffitti we display the data in a div classed simplebox the result is simple:

insert graphic

Write graffittis and see them appearing at the top, once you reach more than 10 only the last 10 graffittis will be displayed.

Adding the graffitti module as a tiki feature.

Once you have a successful module doing everything you want you should add it as a Tiki feature allowing the admin to enable/disable the new feature from the Tiki main screen menu.

In order to do this you must edit tiki-admin.tpl and tiki-admin.php, in tiki-admin.tpl add a checkbox with a name proper to your module example: $feature_graffittis, you can copy-paste-edit the code for another feature. In tiki-admin.php add the code to process the new checkbox, you can copy the code used to process the checkbox for another feature. Once you did this work you will be able to use something like this in tiki-graffitti.php

```
if($feature_graffittis != 'y') {
$smarty->assign('msg',tra("This feature is disabled"));
$smarty->display('error.tpl');
die;
}
```

Note how the error template is called with an error message, note also the use of the tra PHP function to translate the error message string.

There are many imporvements to this module, purging all graffittis, removing graffittis if you are admin, displaying the date along with the graffitti. Etc... Since the module is just an example we ll leave it as is and you can code more if you want.

This was an introduction to the Tiki development world, we strongly hope you like the application and the way in which new features are coded. If you have complaints, suggestions or want to help just contact us.