

CakePHP 4 Site Configuration Using Database



Asyraf Wahi Anuar - May 18, 2020

<https://codethepixel.com/CakePHP-4-Site-Configuration-Using-Database>

This tutorial will show the step to create a site configuration that is accessible from the form interface and stored in a database table: configurations in CakePHP 4. Basically, this site configuration leverage the global variable configuration to make it accessible throughout the system.

Create Database Table: Configurations

Create the database table name configurations which contain the site configuration. You can add or remove any items and change the value according to your requirement. This is the sample of the database table and the value I used in this tutorial.

SQL dump:

```
CREATE TABLE `configurations` (  
  `id` char(36) NOT NULL,  
  `system_name` varchar(255) NOT NULL,  
  `system_abbr` varchar(255) NOT NULL,  
  `organization_name` varchar(255) NOT NULL,  
  `email` varchar(100) NOT NULL,  
  `meta_title` varchar(255) NOT NULL,  
  `meta_keyword` varchar(255) NOT NULL,  
  `meta_desc` varchar(255) NOT NULL,  
  `timezone` varchar(100) NOT NULL,  
  `author` varchar(255) NOT NULL,  
  `user_reg` tinyint(1) NOT NULL,  
  `version` varchar(255) NOT NULL,  
  `created` datetime NOT NULL,  
  `modified` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
INSERT INTO `configurations` (`id`, `system_name`, `system_abbr`, `organization  
( 'myCake4', 'My CakePHP 4', 'MC4', 'Code The Pixel', 'noreply[at]yourDomain.co
```

Load Configuration Model

File Location: ...\\src\\Controller\\AppController.php

Next, load the configuration model into AppController to make it accessible from all controllers.

Then set the global variables for each of the attributes as shown below:

```

public function beforeFilter(\Cake\Event\EventInterface $event)
{
    parent::beforeFilter($event);

    $this->loadModel('Configurations');
    $settings = $this->Configurations->find('all')->first();

    $this->set('system_name', $settings->get('system_name'));
    $this->set('system_abbr', $settings->get('system_abbr'));
    $this->set('organization_name', $settings->get('organization_name'));
    $this->set('email', $settings->get('email'));
    $this->set('meta_title', $settings->get('meta_title'));
    $this->set('meta_keyword', $settings->get('meta_keyword'));
    $this->set('meta_desc', $settings->get('meta_desc'));
    $this->set('timezone', $settings->get('timezone'));
    $this->set('author', $settings->get('author'));
    $this->set('user_reg', $settings->get('user_reg'));
    $this->set('version', $settings->get('version'));
}

```

You can call or print the configuration value from any model in your application (eg: print the system_name value):

```
<?= $system_name; ?>
```

To make it more interesting, bake the configurations table and remove all the others function except edit. Then rename the public function edit to public function setting (obviously it will be in your ConfigurationsController.php). Then set the form to redirect to setting back once updated the value as shown below:

```

public function setting($id = null)
{
    $configuration = $this->Configurations->get($id, [
        'contain' => [],
    ]);
    if ($this->request->is(['patch', 'post', 'put'])) {
        $configuration = $this->Configurations->patchEntity($configuration, $this->request->data());
        if ($this->Configurations->save($configuration)) {
            $this->Flash->success(__('The configuration has been updated.'));

            return $this->redirect(['action' => 'setting', 'myCake4']);
        }
        $this->Flash->error(__('The configuration could not be update. Please, try a'));
    }
    $this->set(compact('configuration'));
}

```

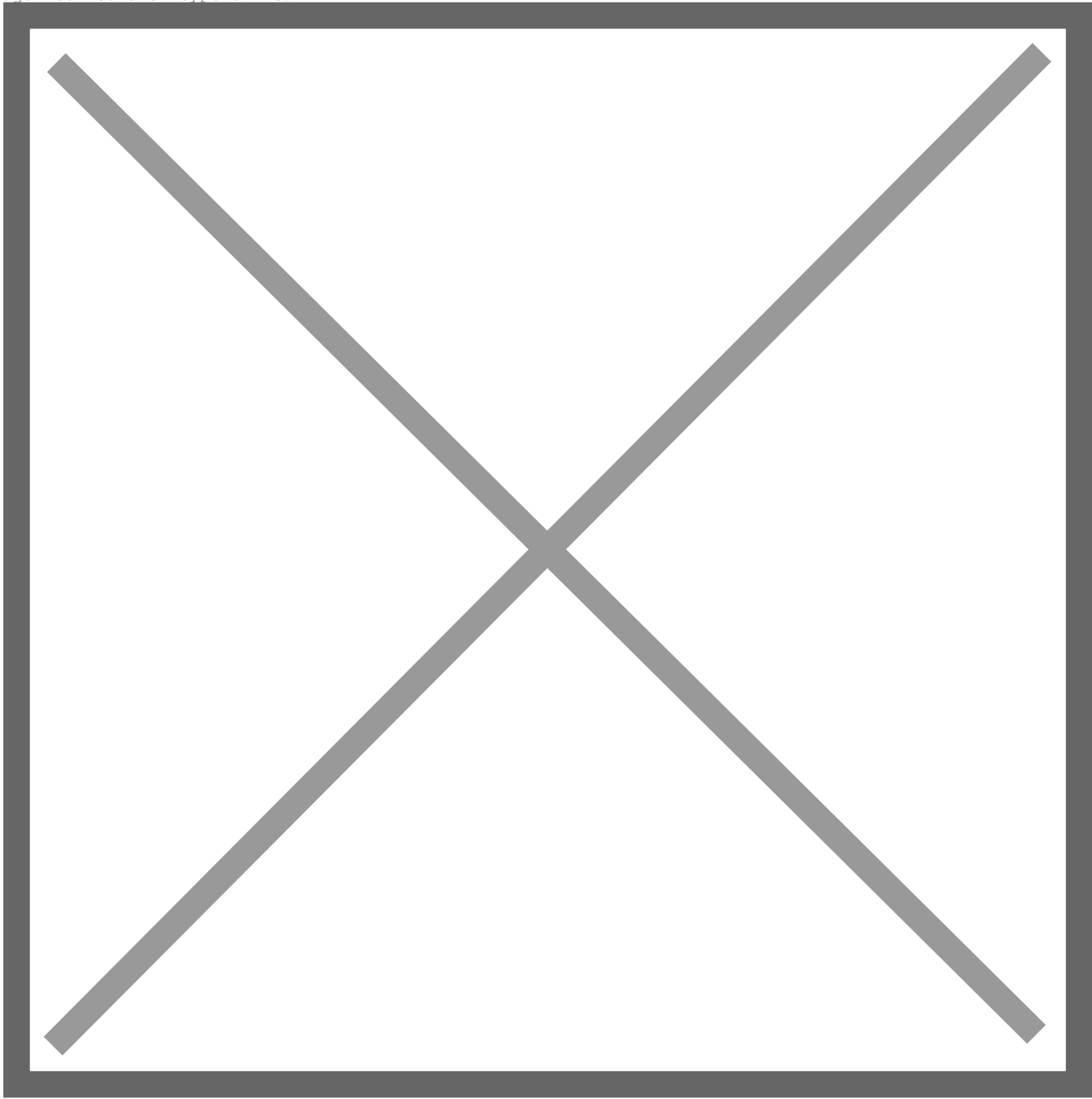
File Location:-...\config\routes.php

Then navigate to routes and add the following codes to manage the redirect. Now you can access your setting /configuration page through localhost/myCake4/config (Note: myCake4 is the ID of

the setting. If you change the value to another value in the database, then use it consistently)

```
$builder->connect('/config/*', ['controller' => 'Configurations', 'action' =>
```

Image not found or type unknown



Note: *The shown image differs from your stock generated view.*

Now you've your own site configuration page which is very useful for other non-programmer or technical people to update the site configuration eg: change the corresponding email, system name etc. That all. Happy coding :)