Authentication

Jill Gundersen

Authentication

- **Editing Content**
  - We don’t want just any user to be able to edit the site’s content.
  - Limit the access to certain URLs based upon if a user is not logged in or not.
- **Authentication**
  - Identifying a user from the provided information (credentials – username and password)
- **Types of Authentication**
  - FormAuthenticate
    - Based on form post data
  - BasicAuthenticate
    - Basic HTTP authentication
  - DigestAuthenticate
    - Digest HTTP authentication

Authentication Setup

- **Add a New Table**
  - Create a new table ‘users’
  - Following the CakePHP naming conventions
- **Bake All**
  - Remember baking the CRUD?
  - Navigate to the Console folder
    - ./cake bake all
    - Choose the users table
- **Add Authentication**
  - Now we are ready to lock down the site.

Components

- **What Are Components?**
  - "Packages of logic that are shared between controllers" – CakePHP.org
- **CakePHP List of Components**
  - Pagination
  - Sessions
    - This has already been implemented in the web site
  - Authentication
  - Security
  - Request Handling
  - Cookie
  - Access Control Lists

AuthComponent

- **AuthComponent**
  - Provides a modular way to identify, authenticate and authorize users
  - Add ‘Auth’ as an element to the $components array
  - Example
    - public $components = array(‘Auth’);

Log In/Log Out

- **Log In View**
  - We need a new form to log in the users.
  - This will entail adding new View and an action in the Users Controller.
- **Auth->login()**
  - Authenticates the user from the log in form, no additional coding needed.
  - Example
    - $this->Auth->login();
- **Log Out**
  - No view needed, but we will need a way to log out the user
- **Auth->logout()**
  - Logs the user out with no additional coding.
  - Example
    - $this->Auth->logout();

beforeFilter()

- **Method: beforeFilter**
  - Used in the controller class, including the AppController
  - Called before any controller action is called
  - This makes it ideal to check user permissions

Auth Allow

- **Auth->allow()**
  - Utilize our new beforeFilter method to allow the user access to certain pages/actions.
  - By default CakePHP will deny access to all pages if the user is not logged in.
  - Grant Permissions
    - Specify the actions in the controllers that are allowed to be accessed if the user is not logged in
  - Can be added site wide (AppController) or inside each of the controllers
  - Example
    - $this->Auth->allow(‘index’, ‘view;’);

Is User Logged In?

- **Limit Certain Content if a User is Not Logged In**
  - In order to check if a user is logged in we call the method user() from the AuthComponent class
  - Example
    - AuthComponent::user();

Passwords

- **Field Password Compared**
  - The Auth->login() method compares the log in post data (hashed) to the table field data
  - Stored password for this reason need to be saved in their hashed state
- **How Do We Accomplish This?**
  - We need to be able to intercept our post data before it gets saved to the database
  - CakePHP provides a nifty function called beforeSave()
    - This allows up to manipulate the data before sending it on to be saved by the Model
- **Location**
  - Since the Models in CakePHP deal with the data, this method is used in the Model class

Password Hashing

- **Password Variable**
  - All post data is stored in the data variable with a key based on the Model name:
    - $this->data['ModelName']
  - The field in the database is the next key in the data associated array
  - Example
    - $this->data['User']['password']
- **AuthComponent Provides Hashing Method**
  - AuthComponent::password()
  - Example
    - AuthComponent::password($this->data['User']['password']);

Summary

- **Create Users Table**
  - Baked the CRUD
- **Created Login Page**
- **Added the Auth Component**
- **Limited Pages**
- **Hash and Store Passwords**

YUM!