Search

- About
- Portfolio
- Coding
- Blog
- Contact

Simple ACL Class for CodeIgniter

Hey guys, I was originally using an ACL Class that was posted on net tuts. After scouring the internet for an ACL plugin that was similar to the net tuts one I opted to just adapt the net tuts one since I couldn’t find one similar. I tried out the Zend ACL library but it didnt (that i saw) have the same functionality

- User by user permissions
- Multiple Roles
- Database configured
- Etc.

Here is the net.tutsplus.com walk through if anyone needs it.
http://net.tutsplus.com/tutorials/php/a-better-login-system/

Database
- Same as the net tuts, just renamed the tables for my own preference.
- USER_DATA can be basically anything you want, just needs to have “id

```sql
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Table structure for table `perm_data`
--

CREATE TABLE `perm_data` (
  `ID` bigint(20) unsigned NOT NULL auto_increment,
  `permKey` varchar(30) NOT NULL,
  `permName` varchar(30) NOT NULL,
  PRIMARY KEY  (`ID`),
  UNIQUE KEY `permKey` (`permKey`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=39 ;

```

```
-- --------------------------------------------------------

--
-- Table structure for table `role_data`
--

CREATE TABLE `role_data` (
  `ID` bigint(20) unsigned NOT NULL auto_increment,
  `roleName` varchar(20) NOT NULL,
  PRIMARY KEY  (`ID`),
  UNIQUE KEY `roleName` (`roleName`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

-- --------------------------------------------------------

--
-- Table structure for table `role_perms`
--

CREATE TABLE `role_perms` (
  `ID` bigint(20) unsigned NOT NULL auto_increment,
  `roleID` bigint(20) NOT NULL,
  `permID` bigint(20) NOT NULL,
  `value` tinyint(1) NOT NULL default '0',
  `addDate` datetime NOT NULL,
  PRIMARY KEY  (`ID`),
  UNIQUE KEY `roleID_2` (`roleID`,`permID`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- --------------------------------------------------------

--
-- Table structure for table `user_data`
--

CREATE TABLE `user_data` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `username` varchar(20) NOT NULL,
  `password` text NOT NULL,
  `name` tinytext NOT NULL,
  `address` tinytext,
  `address2` tinytext,
  `city` tinytext,
  `state` tinytext,
  `zip` tinytext,
  `cellphone` tinytext,
  `telephone` tinytext,
  `email` tinytext,
  `aim` tinytext,
  `yahoo` tinytext,
  `icq` tinytext,
  `other` tinytext,
  `dateAdded` timestamp NOT NULL default CURRENT_TIMESTAMP,
  PRIMARY KEY  (`ID`),
  KEY `Username` (`username`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- --------------------------------------------------------

--
-- Table structure for table `user_perms`
--

CREATE TABLE `user_perms` (
  `ID` bigint(20) unsigned NOT NULL auto_increment,
  `userID` bigint(20) NOT NULL,
  `permID` bigint(20) NOT NULL,
  `value` tinyint(1) NOT NULL default '0',
  `addDate` datetime NOT NULL,
  PRIMARY KEY  (`ID`),
  UNIQUE KEY `userID` (`userID`,`permID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- --------------------------------------------------------

--
-- Table structure for table `user_roles`
--

CREATE TABLE `user_roles` (
  `userID` bigint(20) NOT NULL,
  `roleID` bigint(20) NOT NULL,
  `addDate` datetime NOT NULL,
  UNIQUE KEY `userID` (`userID`,`roleID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

The Class File
- Just a converted version of the net tuts class.

```
class acl
{
    var $perms = array();       //Array : Stores the permissions for the user
    var $userID;            //Integer : Stores the ID of the current user
    var $userRoles = array();   //Array : Stores the roles of the current user
    var $ci;
    function __construct($config=array()) {
        $this->ci = &get_instance();

        $this->userID = floatval($config['userID']);
        $this->userRoles = $this->getUserRoles();
        $this->buildACL();
    }

    function buildACL() {
        //first, get the rules for the user's role
        if (count($this->userRoles) > 0)
        {
            $this->perms = array_merge($this->perms,$this->getRolePerms($this->userRoles));
        }
        //then, get the individual user permissions
        $this->perms = array_merge($this->perms,$this->getUserPerms($this->userID));
    }

    function getPermKeyFromID($permID) {
        //$strSQL = "SELECT `permKey` FROM `".DB_PREFIX."permissions` WHERE `ID` = " . floatval($permID) . " LIMIT 1";
        $this->ci->db->select('permKey');
        $this->ci->db->where('id',floatval($permID));
        $sql = $this->ci->db->get('perm_data',1);
        $data = $sql->result();
        return $data[0]->permKey;
    }

    function getPermNameFromID($permID) {
        //$strSQL = "SELECT `permName` FROM `".DB_PREFIX."permissions` WHERE `ID` = " . floatval($permID) . " LIMIT 1";
        $this->ci->db->select('permName');
        $this->ci->db->where('id',floatval($permID));
        $sql = $this->ci->db->get('perm_data',1);
        $data = $sql->result();
        return $data[0]->permName;
    }

    function getRoleNameFromID($roleID) {
        //$strSQL = "SELECT `roleName` FROM `".DB_PREFIX."roles` WHERE `ID` = " . floatval($roleID) . " LIMIT 1";
        $this->ci->db->select('roleName');
        $this->ci->db->where('id',floatval($roleID),1);
        $sql = $this->ci->db->get('role_data');
        $data = $sql->result();
        return $data[0]->roleName;
    }

    function getUserRoles() {
        //$strSQL = "SELECT * FROM `".DB_PREFIX."user_roles` WHERE `userID` = " . floatval($this->userID) . " ORDER BY `addDate` ASC";

        $this->ci->db->where(array('userID'=>floatval($this->userID)));
        $this->ci->db->order_by('addDate','asc');
        $sql = $this->ci->db->get('user_roles');
        $data = $sql->result();

        $resp = array();
        foreach( $data as $row )
        {
            $resp[] = $row->roleID;
        }
        return $resp;
    }

    function getAllRoles($format='ids') {
        $format = strtolower($format);
        //$strSQL = "SELECT * FROM `".DB_PREFIX."roles` ORDER BY `roleName` ASC";
        $this->ci->db->order_by('roleName','asc');
        $sql = $this->ci->db->get('role_data');
        $data = $sql->result();

        $resp = array();
        foreach( $data as $row )
        {
            if ($format == 'full')
            {
                $resp[] = array("id" => $row->ID,"name" => $row->roleName);
            } else {
                $resp[] = $row->ID;
            }
        }
        return $resp;
    }

    function getAllPerms($format='ids') {
        $format = strtolower($format);
        //$strSQL = "SELECT * FROM `".DB_PREFIX."permissions` ORDER BY `permKey` ASC";

        $this->ci->db->order_by('permKey','asc');
        $sql = $this->ci->db->get('perm_data');
        $data = $sql->result();

        $resp = array();
        foreach( $data as $row )
        {
            if ($format == 'full')
            {
                $resp[$row->permKey] = array('id' => $row->ID, 'name' => $row->permName, 'key' => $row->permKey);
            } else {
                $resp[] = $row->ID;
            }
        }
        return $resp;
    }

    function getRolePerms($role) {
        if (is_array($role))
        {
            //$roleSQL = "SELECT * FROM `".DB_PREFIX."role_perms` WHERE `roleID` IN (" . implode(",",$role) . ") ORDER BY `ID` ASC";
            $this->ci->db->where_in('roleID',$role);
        } else {
            //$roleSQL = "SELECT * FROM `".DB_PREFIX."role_perms` WHERE `roleID` = " . floatval($role) . " ORDER BY `ID` ASC";
            $this->ci->db->where(array('roleID'=>floatval($role)));

        }
        $this->ci->db->order_by('id','asc');
        $sql = $this->ci->db->get('role_perms'); //$this->db->select($roleSQL);
        $data = $sql->result();
        $perms = array();
        foreach( $data as $row )
        {
            $pK = strtolower($this->getPermKeyFromID($row->permID));
            if ($pK == '') { continue; }
            if ($row->value === '1') {
                $hP = true;
            } else {
                $hP = false;
            }
            $perms[$pK] = array('perm' => $pK,'inheritted' => true,'value' => $hP,'name' => $this->getPermNameFromID($row->permID),'id' => $row->permID);
        }
        return $perms;
    }

    function getUserPerms($userID) {
        //$strSQL = "SELECT * FROM `".DB_PREFIX."user_perms` WHERE `userID` = " . floatval($userID) . " ORDER BY `addDate` ASC";

        $this->ci->db->where('userID',floatval($userID));
        $this->ci->db->order_by('addDate','asc');
        $sql = $this->ci->db->get('user_perms');
        $data = $sql->result();

        $perms = array();
        foreach( $data as $row )
        {
            $pK = strtolower($this->getPermKeyFromID($row->permID));
            if ($pK == '') { continue; }
            if ($row->value == '1') {
                $hP = true;
            } else {
                $hP = false;
            }
            $perms[$pK] = array('perm' => $pK,'inheritted' => false,'value' => $hP,'name' => $this->getPermNameFromID($row->permID),'id' => $row->permID);
        }
        return $perms;
    }

    function hasRole($roleID) {
        foreach($this->userRoles as $k => $v)
        {
            if (floatval($v) === floatval($roleID))
            {
                return true;
            }
        }
        return false;
    }

    function hasPermission($permKey) {
        $permKey = strtolower($permKey);
        if (array_key_exists($permKey,$this->perms))
        {
            if ($this->perms[$permKey]['value'] === '1' || $this->perms[$permKey]['value'] === true)
            {
                return true;
            } else {
                return false;
            }
        } else {
            return false;
        }
    }
}
```

Example Usage

```
// Use whatever user script you would like, just make sure it has an ID field to tie into the ACL with
$this->load->library('user',array('username'=>'admin','password'=>'abc123'));

// Get the user's ID and add it to the config array
$config = array('userID'=>$this->user->getUserID());

// Load the ACL library and pas it the config array
$this->load->library('acl',$config);

// Get the perm key
// I'm using the URI to keep this pretty simple ( http://www.example.com/test/this ) would be 'test_this'
$acl_test = $this->uri->segment(1).'_';
$acl_test .= ($this->uri->segment(2)!="")?$this->uri->segment(2):'view';

// If the user does not have permission either in 'user_perms' or 'role_perms' redirect to login, or restricted, etc
if ( !$this->acl->hasPermission($acl_test) ) {
    redirect('/login/');
}
```

Installation

First, I have these autoloaded with the file /application/config/autoload.php

- **Libraries** database and session
- **Helpers** url

Then I placed the ACL Class in a file named acl.php and put it in /application/libraries/ and loaded like in the example above. All the acl class needs is a user ID that corresponds to the user ID in the database (note the table "user_perms" and "user_roles") **USER_DATA: is my own custom users database, it is only there for convenience** you can use whatever user data table you would like. Just make sure the USERDATA.id matches with the user_roles.userID and users_perms.userID

Once you give the class constructor an userID it just tells you if that user has the permission. **NOTE:** You must add permissions to the "perm_data" table and link users to it

22 Thoughts

1. *Nishan Karassik* says:
   March 30, 2011 at 10:52 pm

   Thanks for putting this together. I looked an looked and didn't like Zend Acl either.

   I have made a few adustments and will make a few more as I dig a little deeper on a new project, but I thought I would throw these out there.

   The hasRole function bothered me because it needed the roleid instead of roleName. I had to change getUserRoles and the hasRole functions as follows:

```
function getUserRoles() {
//$strSQL = "SELECT UR.roleid, LOWER(RD.roleName) roleName FROM
`".DB_PREFIX."user_roles` UR
//INNER JOIN role_data RD ON RD.id = UR.roleid WHERE `userid` = " . floatval($this->userid) . "
//ORDER BY `addDate` ASC";
$this->ci->db->select('UR.roleid, LOWER(RD.roleName) roleName');
$this->ci->db->from('user_roles UR');
$this->ci->db->join('role_data RD', 'RD.id = UR.roleid');
$this->ci->db->where(array('userid'=>floatval($this->userid)));
$this->ci->db->order_by('addDate','asc');
$sql = $this->ci->db->get();
$data = $sql->result();

$resp = array();
foreach( $data as $row )
{
$resp[$row->roleName] = $row->roleid;
}
return $resp;
}
```

   AND

```
function hasRole($roleName) {
$roleName = strtolower($roleName);
foreach($this->userRoles as $k => $v)
{
if ($k === $roleName)
{
return true;
}
}
return false;
}
```

I will likely do similar adjustments to the getRolePerms($role) to instead use roleName when I start developing the admin panel for this.

Also I’ve changed the ‘value’ field in both tables and appropriate functions to ‘active’. If I’m understanding the design it’s more descriptive.

Reply

2. *ken* says:
   April 20, 2011 at 5:20 am

   how can i intergrate this on a existing login system… fairly new to PHP…

   Reply

   ○ *Brian* says:
     May 2, 2011 at 1:44 pm

     Is your existing login system based on the Code Igniter framework?

     If not, this script will be pretty difficult to translate it over.

     read this link, it is a really well written walk through of the script.

     http://net.tutsplus.com/tutorials/php/a-better-login-system/

     Reply

3. *JonoB* says:
   June 6, 2011 at 5:22 am

   Brian,

   This is great, thank you. I am looking to implement ACL in a CI project, and was looking at Zend_Acl, until I stumbled across your website. As a matter of interest, what does the Zend implementation miss out on that yours does provide?

   Reply

   ○ *Brian* says:
     June 6, 2011 at 6:50 am

     For me, it was a need for:
     - User by user permissions

- Multiple Roles
- Database configured

While the Zend ACL may offer these, I didn’t have the time to look through countless tutorials etc looking for something I already “had”.

It also came down to an easy of use issue. This class is pretty much very straight forward when it comes to setting it up and using it.

[Reply](#)

4. *Pedma* says:
   [July 18, 2011 at 4:33 am](#)

   Hi,

   I’m new in CodeIgniter.
   Could you please give a real example how to use this ACL ?
   Is it possible to use this for ‘simple login’ (http://codeigniter.com/wiki/Simplelogin/) ?

   Thanks
   -pedma-

   [Reply](#)

5. *pico* says:
   [November 2, 2011 at 12:03 am](#)

   Hey, thanks for you tutorial. But i hope to see with a controller to use this libraries. If it possible let’s me know coz of i’m newbie in CI. I have no idea how to use it. But still hope this tutorial will help me alot

   [Reply](#)

   ○ *Brian* says:
     [January 10, 2012 at 6:31 am](#)

     The example given above, basically is a real example in a controller. I normally have that bit of code in the constructor of the page controller. It’s been a while since I’ve used CI (or php for that matter)

     [Reply](#)

6. *Felipe Castro* says:
   [November 28, 2011 at 6:23 pm](#)

   Beautiful script for codeigniter. I’ve made a few adjustments that I post bellow.

   I needed to be able to configure if the user had permission to either Create, Read, Delete, or all of them.

   class acl {
   /* Actions::::

```
* Create 1
* Read 2
* Update 4
* Delete 8
* The allowance is made by a sum of the actions allowed.
* Ex.: user can read and update (2+4)=6 … so ill put 6 instead of 1 or 0.
*
* if(!$this->acl->hasPermission(‘entries_complete_access’)) {
echo “No no”;
} else
* echo “yeah”;
}
*
*
*/

var $perms = array(); //Array : Stores the permissions for the user
var $userID; //Integer : Stores the ID of the current user
var $userRoles = array(); //Array : Stores the roles of the current user
var $ci;

function __construct($config=array()) {
$this->ci = &get_instance();
$this->userID = floatval($this->ci->session->userdata(‘account_id’));
$this->userRoles = $this->getUserRoles();
$this->buildACL();
}

function buildACL() {
//first, get the rules for the user’s role
if (count($this->userRoles) > 0) {
$this->perms = array_merge($this->perms, $this->getRolePerms($this->userRoles));
}
//then, get the individual user permissions
$this->perms = array_merge($this->perms, $this->getUserPerms($this->userID));

}

function getPermKeyFromID($permID) {
//$strSQL = “SELECT `permKey` FROM `”.DB_PREFIX.”permissions` WHERE `ID` = ” .
floatval($permID) . ” LIMIT 1″;
$this->ci->db->select(‘permKey’);
$this->ci->db->where(‘id’, floatval($permID));
$sql = $this->ci->db->get(‘perm_data’, 1);
$data = $sql->result();
return $data[0]->permKey;
}

function getPermNameFromID($permID) {
//$strSQL = “SELECT `permName` FROM `”.DB_PREFIX.”permissions` WHERE `ID` = ” .
floatval($permID) . ” LIMIT 1″;
$this->ci->db->select(‘permName’);
```

```
$this->ci->db->where(‘id’, floatval($permID));
$sql = $this->ci->db->get(‘perm_data’, 1);
$data = $sql->result();
return $data[0]->permName;
}

function getRoleNameFromID($roleID) {
//$strSQL = “SELECT `roleName` FROM `”.DB_PREFIX.”roles` WHERE `ID` = ” . floatval($roleID)
. ” LIMIT 1″;
$this->ci->db->select(‘roleName’);
$this->ci->db->where(‘id’, floatval($roleID), 1);
$sql = $this->ci->db->get(‘role_data’);
$data = $sql->result();
return $data[0]->roleName;
}

function getUserRoles() {
//$strSQL = “SELECT * FROM `”.DB_PREFIX.”user_roles` WHERE `userID` = ” .
floatval($this->userID) . ” ORDER BY `addDate` ASC”;

$this->ci->db->where(array(‘userID’ => floatval($this->userID)));
$this->ci->db->order_by(‘addDate’, ‘asc’);
$sql = $this->ci->db->get(‘user_roles’);
$data = $sql->result();

$resp = array();
foreach ($data as $row) {

$resp[] = $row->roleID;
}
return $resp;
}

function getAllRoles($format=’ids’) {
$format = strtolower($format);
//$strSQL = “SELECT * FROM `”.DB_PREFIX.”roles` ORDER BY `roleName` ASC”;
$this->ci->db->order_by(‘roleName’, ‘asc’);
$sql = $this->ci->db->get(‘role_data’);
$data = $sql->result();

$resp = array();
foreach ($data as $row) {
if ($format == ‘full’) {
$resp[] = array(“id” => $row->ID, “name” => $row->roleName);
} else {
$resp[] = $row->ID;
}
}
return $resp;
}

function getAllPerms($format=’ids’) {
```

```
$format = strtolower($format);
//$strSQL = “SELECT * FROM `”.DB_PREFIX.”permissions` ORDER BY `permKey` ASC”;

$this->ci->db->order_by(‘permKey’, ‘asc’);
$sql = $this->ci->db->get(‘perm_data’);
$data = $sql->result();

$resp = array();
foreach ($data as $row) {
if ($format == ‘full’) {
$resp[$row->permKey] = array(‘id’ => $row->ID, ‘name’ => $row->permName, ‘key’ =>
$row->permKey);
} else {
$resp[] = $row->ID;
}
}
return $resp;
}

function getRolePerms($role) {
if (is_array($role)) {
//$roleSQL = “SELECT * FROM `”.DB_PREFIX.”role_perms` WHERE `roleID` IN (” .
implode(“,”,$role) . “) ORDER BY `ID` ASC”;
$this->ci->db->where_in(‘roleID’, $role);
} else {
//$roleSQL = “SELECT * FROM `”.DB_PREFIX.”role_perms` WHERE `roleID` = ” . floatval($role) .
” ORDER BY `ID` ASC”;
$this->ci->db->where(array(‘roleID’ => floatval($role)));
}
$this->ci->db->order_by(‘id’, ‘asc’);
$sql = $this->ci->db->get(‘role_perms’); //$this->db->select($roleSQL);
$data = $sql->result();
$perms = array();
foreach ($data as $row) {
$pK = strtolower($this->getPermKeyFromID($row->permID));

if ($pK == ”) {
continue;
}
/*if ($row->value == ’1′) {
$hP = true;
} else {
$hP = false;
}*/
if ($row->value == ’0′) {
$hP = false;
} else {
$hP = $row->value ;
}

$perms[$pK] = array(‘perm’ => $pK, ’1inheritted’ => true, ‘value’ => $hP, ‘name’ =>
$this->getPermNameFromID($row->permID), ‘id’ => $row->permID);
}
return $perms;
}

function getUserPerms($userID) {
//$strSQL = “SELECT * FROM `”.DB_PREFIX.”user_perms` WHERE `userID` = ” .
floatval($userID) . ” ORDER BY `addDate` ASC”;

$this->ci->db->where(‘userID’, floatval($userID));
$this->ci->db->order_by(‘addDate’, ‘asc’);
$sql = $this->ci->db->get(‘user_perms’);
$data = $sql->result();

$perms = array();
foreach ($data as $row) {
$pK = strtolower($this->getPermKeyFromID($row->permID));
if ($pK == ”) {
continue;
}
/*if ($row->value == ’1′) {
$hP = true;
} else {
$hP = false;
}*/
if ($row->value == ’0′) {
$hP = false;
} else {
$hP = $row->value ;
}

$perms[$pK] = array(‘perm’ => $pK, ’2inheritted’ => false, ‘value’ => $hP, ‘name’ =>
$this->getPermNameFromID($row->permID), ‘id’ => $row->permID);
}
return $perms;
}

function hasRole($roleID) {
foreach ($this->userRoles as $k => $v) {
if (floatval($v) === floatval($roleID)) {
return true;
}
}
return false;
}

function actionPerm($value, $wanted) {
/* Actions::::
* Create 1
* Read, 2
* Update, 4
* Delete 8
*/
```

```
$action['create'] = array(’1′, ’3′, ’5′, ’9′, ’11′, ’13′, ’15′); //1
$action['read'] = array(’2′, ’3′, ’6′, ’10′, ’14′, ’15′); //2
$action['update'] = array(’4′, ’5′, ’6′, ’7′, ’12′, ’13′, ’14′, ’15′); //4
$action['delete'] = array(’8′, ’9′, ’10′, ’11′, ’12′, ’13′, ’14′, ’15′); //8
$action['all'] = array(’15′);

if (in_array($value, $action[$wanted], true)) {
return true;
} else {
return false;
}
}

function hasPermission($permKey, $action = ‘all’) {

$permKey = strtolower($permKey);

if (array_key_exists($permKey, $this->perms)) {
if ($this->actionPerm($this->perms[$permKey]['value'], $action)) {

return true;
} else {
return false;
}
} else {
return false;
}
/* OLD METHOD
if ($this->perms[$permKey]['value'] === ’1′ || $this->perms[$permKey]['value'] === true)
{
return true;
} else {
return false;
}
} else {
return false;
}
*/
}

}
```

Reply

7. Joe says:
March 16, 2012 at 5:52 pm

Does any one have some sample data that I can add in, I don’t know how to use this. and does some one have a sample of a admin section that we can use to edit the settings from the tables.

Reply

8. *Andrew* says:
   March 17, 2012 at 4:25 am

   Hey there,

   Excellent approach. I have only one question, why are you checking everything against sql every time. Should be better to check once upon login and then store at least the user permission in session. In fact I would store everything in session and then access it from there rather than querying the database couple of times before each action.

   What do you think?

   Reply

   - *Brian* says:
     March 29, 2012 at 7:43 am

     I wouldn’t disagree that there are unnecessary db calls, but it is literally a straight conversion of the ACL that was posted on net tuts.

     Reply

9. *Fons* says:
   March 28, 2012 at 11:38 pm

   Thank you for this plugin, i’m implementing it as we speak. Now I have put your example usage in my login method in my authentication controller.. how do I repeat this for other controllers in the most efficient way (keeping code duplication the lowest as possible)

   thanks again!

   Reply

   - *Fons* says:
     March 29, 2012 at 6:19 am

     I have fixed this issue, I have created a function in MY_Controller (in application/core). I call it in the constructor of my controllers to check if the user has the right permissions and it works. Thanks alot for this usefull script! You saved me 😃

     Reply

     - *Brian* says:
       March 29, 2012 at 7:42 am

       You may also be able to create a class that is auto-loaded or you classes extend that automatically checks the auth on the page.

       Reply

10. *Fons* says:
    April 10, 2012 at 1:36 am

Thanks again for this awesome library! I have succesfully implemented it in my project. One quick question: i'm not going to use perms based on users only on roles, can I just delete the db table for user_perms and the code in the library or will that fonk things up?

Reply

- *Fons* says:
  April 10, 2012 at 5:24 am

  Nevermind, allready fixed this 😉

  Reply

11. *Kumar* says:
    April 16, 2012 at 9:25 pm

    How can I do for the following situation.

    in the page list of users, admin should have the options create, edit, delete and view whereas normal user should have only view permission.

    Please let me know best way to implement it

    Reply

12. *Troy* says:
    May 5, 2012 at 4:40 pm

    How are you editing the permissions and roles. Have you created an interface similar to the tutorial. If so how have you managed to call an instance of the library so that the ACL is built with the user to be edited details. I am trying to build an admin section for editing the user permissions but keep tripping over my own feet.

    Reply

    - *Brian* says:
      August 27, 2012 at 11:37 am

      Yes, I've created a few different interfaces for the couple of projects I've used this in. Currently they are all either private or too big to post an example.

      The net tuts link has a pretty good walk through of setting up an interface that would only need minor tweaking to work with code igniter.

      Reply

13. *Arif* says:
    July 9, 2012 at 4:44 am

    Hi Brian,

    Thanks for this simple class. This code really worked for me. I have succesfully implemented it in my project.

Reply

14. *Dollardad* says:
August 3, 2012 at 10:02 pm

I’ve created a Role Base ACL Class and Login system for Codeigniter (Version 2.1.2) that does not require a database table.

Simple system for small websites (like blog or small business) that you need login for just a few users.

You can download it either on my personal blog
http://www.kevinphillips.co.nz
or github
https://github.com/dollardad/CI-ACL

Reply

Your Thoughts...

Name (required)

Mail (will not be published) (required)

Website

Submit

Similar Projects

- Simple File Caching Script

  On a recent project, I needed a quick, easy and simple caching function for XML files, so whipped up this little function.

  It takes:

  - The URL of the file / XML you need to cache
  - The cache file, if one is not specified, it creates and uses “temp.txt”
  - The amount of time to cache the file, if one is not specified, it uses 12 minutes (you can change this to whatever you would like

- Quick and Easy Modal Window with CSS3

I’ve been playing around on jsfiddle.net and whipped up a really stylish modal box using only CSS3, no images, etc. The icon can be changed out to any icon you want, especially if you stick with Pure CSS GUI Icons.

- Elegant Curved HR with CSS3

  Here is some simple CSS3 to create a clean and elegant HR, it does not necessarily degrade the best, but its not too bad. This was mainly an exercise in CSS3 for me. So lets get to it. The html, just a basic <hr> element with a class of (whatever you’d like) “oval” <hr class=”oval” [...]

- Simple ACL Class for CodeIgniter

  Hey guys, I was originally using an ACL Class that was posted on net tuts. After scouring the internet for an ACL plugin that was similar to the net tuts one I opted to just adapt the net tuts one since I couldn’t find one similar. I tried out the Zend ACL library but it didnt (that i saw) have the same functionality

  - User by user permissions
  - Multiple Roles
  - Database configured
  - Etc.

Categories

- Blog
- Coding
- Portfolio
  - Photography
  - Print
  - Web

Tags

Aqua Sonix bradford pear cache Cars code igniter Continental CSS CSS3 denton elegant events fountain frisco hr html html5 Illustrator Invite lake lewisville Lincoln little elm Metro modal Muscle Photography Php Portrait Print script simple Sunrise Website window Wok Yellow 7

Recent Posts

- Quick and Easy Modal Window with CSS3
- Elegant Curved HR with CSS3
- Self Portrait
- Aqua Sonix Ads
- Aqua Sonix Branding

Subscribe to RSS Feed – Posts or just Comments