

Yii-Rights documentation

Table of Contents

Introduction	2
Requirements	2
Installation	2
Configuration	3
Installer	3
Accessing Rights	4
Routes	4
Permissions	4
Operations, Tasks and Roles	4
Create Auth Item	4
Assignments	4
User Assignments	4
Automatic Controller Access Control (ACAC)	5
Internationalization (I18N)	6
Translating Rights	6
Module file Structure	7
Change Log	8
Glossary	9

Introduction

Rights is a module for Yii web development framework (<http://www.yiiframework.com>). With Rights you can easily manage user permissions within your Yii application. This extension utilizes Yii's built-in database Authorization Manager (CDbAuthManager) to provide an extensive web interface for advanced access control.

Project is hosted on Google Code at: <http://code.google.com/p/yii-rights>

Requirements

Rights require your application to have a working database connection and a User model with an id and name attribute. This is because CDbAuthManager requires a database connection and its Assignment component assigns items directly to users. If you created your application using the Yii console application (yiic) you do not need to worry about this because your application already has a compatible User model.

Installation

1. Download the module at <http://code.google.com/p/yii-rights/downloads/list>.

Tip: You can also do a checkout. Instructions at: <http://code.google.com/p/yii-rights/source/checkout>.

2. Extract the module to your application's **protected/modules/rights** folder.

Tip: The **modules** folder might not exist, if so create it.

3. Add the following lines to your application's configuration:

```
'import'=>array(
    ...
    'application.modules.rights.components.RightsWebUser',
),
...
'components'=>array(
    ...
    'user'=>array(
        'class'=>'RightsWebUser',
        ...
    ),
    'authManager'=>array(
        'class'=>'CDbAuthManager',
        'connectionID'=>'db',
    ),
    ...
),
...
'modules'=>array(
    'rights'=>array(
        'install'=>true, // Add this line only if you wish to use the installer.
    ),
),
```

Do not include the dots. They are merely placeholders for additional content in the configuration file.

Configuration

```
'rights'=>array(
    'superUserRole'=>'Admin',           // Name of the role with super user privileges.
    'defaultRoles'=>array('Guest'),    // List of role names that are assigned to all users.
    'superUsers'=>array(1=>'admin'),    // List of users with access to Rights and elsewhere.
    'userClass'=>'User',               // User model class name.
    'usernameColumn'=>'username',       // Name of the user name column in the database.
    'enableBizRule'=>true,              // Whether to enable auth item business rules.
    'enableBizRuleData'=>false,         // Whether to enable data for business rules.
    'install'=>false,                  // Whether the module should be installed on first run.
    'layout'=>'application.views.layouts.column1', // Layout to use for displaying Rights.
    'css'=>'rights.css',               // Style sheet to use for Rights.
),
```

Values used in this example are the default values.

Note: In case your User model is not called 'User' or if you username column is not called 'username' you need to override those by setting them in the module configuration.

Installer

You can easily install Rights using its installer. This can be done by setting the **install** configuration to **true** (see example above). When the module is loaded the installer will check if the tables required by CDbAuthManager exist and create them if necessary. The installer will also insert necessary Authorization Items and Assignments according to your configuration.

Note: If you do not disable install after the module has been installed it will check if it's installed every single time the module is run. This is why you should set the 'install' configuration to false once the module is installed.

Tip: If you do not wish to use the installer you can also run the module's database schema manually. It can be found under the modules data folder.

Accessing Rights

Once Rights is installed you can access it from:

```
path/to/application/index.php?r=rights
```

Tip: If the main layout is not displaying for Rights go to your **column1.php** (and **column2.php**) in the applications views/layouts folder and make sure that the view path given to beginContent-method is a full path ('**application.views.layouts.main**') instead of only the name of the layout ('**main**').

Routes

Here is a full list of Rights' routes:

```
rights/main/permissions    // Permissions overview. (Default action)
rights/main/operations    // List of operations.
rights/main/tasks         // List of tasks.
rights/main/roles         // List of roles.
rights/authItem/create    // Create new Auth Items (operations, tasks and roles). (Default action)
rights/authItem/update    // Update and add/remove children to existing Auth Items.
rights/assignment/view    // Assignments overview. (Default action)
rights/assignment/user    // View assignments and add/remove assignments for a specific user.
```

Permissions

Under Permissions you can easily assign operations and tasks to your roles and revoke operations and tasks from your roles. In addition you can see which permissions each role has and if they are directly assigned or inherited. If they are inherited you can also see from where they are inherited.

Operations, Tasks and Roles

Under these you can easily manage each type of Auth Item separately. Here you can see a list of Auth Items which can be updated or deleted. In addition you can also see how many children each item has.

Note: *Pagination is not available at this time because CDbAuthManager uses direct queries to fetch auth items. It would require quite a lot of work to allow pagination for these.*

Create Auth Item

Here you can create new operations, tasks and roles. Keep in mind that the type cannot be changed afterward.

Assignments

Under Assignments you can see which Assignments each user has.

User Assignments

Here you can view a specific user's Assignments. You can also add Assignments to or remove Assignments from that user.

Automatic Controller Access Control (ACAC)

Rights allows for automatic controller access control by implementing an own filter, RightsFilter. In order to use this filter you need to extend your application base controller (normally called Controller) from the RightsBaseController.

When you have done this you can add **rights** to your controller's filters and optionally you can also tell it which actions should always be allowed by implementing the `allowedActions`-method.

```
public function filters()
{
    return array(
        'rights',
    );
}

public function allowedActions()
{
    return 'index, update, suggestedTags';
}
```

Example implementation

You need to create an operation for each controller action that is not always allowed or they will be denied from everyone except super users. The naming policy is the following:

```
(Module_)Controller_Action
```

After creating the necessary operations access will automatically be checked before running the action.

Note: The reason why Rights uses underscores is improve the readability and to allow for easier beautification. In the views the names are beautified by replacing the underscores with spaces.

Internationalization (I18N)

Rights support internationalization. The module will automatically use the same language as your application if a translation is available. To set your application's language to e.g. Brazilian Portuguese add the following line to your application's configuration:

```
'language'=>'pt_br' // CountryId(_RegionId).
```

Translating Rights

If you wish to translate Rights use the template for translations (found under messages/templates). When the translation is ready place it under messages/CountryId(_RegionId) in a file called tr.php.

If for example you would like to translate the module into Spanish you would place the translation under *messages/es* and if you would also like to translate the module into Brazilian Portuguese you would place that translation under *messages/pt_br*.

Tip: Contact me if you need help translating the module or wish to share your translation with the community. cniska@live.com

File structure

Here is the complete file structure of the module with short descriptions for what each file is used for:

```
rights
  assets
    rights.css           // Default style sheet.
    rights.js            // JavaScript.
  components
    Rights.php           // Provides access to Rights from outside the module.
    RightsAuthorizer.php // Provides the module's core functionality.
    RightsBaseController.php // Base Controller with automatic controller access control.
    RightsFilter.php     // Filter for automatic controller access control (ACAC).
    RightsInstaller.php  // Installer creates the necessary db tables.
    RightsWebUser.php    // Web user with support for super users.
  controllers
    AssignmentController.php // Handles assignment actions.
    AuthItemController.php  // Handles auth item actions.
    MainController.php      // Handles core actions.
  data
    schema.sql            // Database schema.
  messages
    template
      tr.php             // Translation template.
  models
    AssignmentForm.php    // Form model for assigning items to users.
    AuthChildForm.php     // Form model for adding children to auth items.
    AuthItemForm.php     // Form model for creating/updating auth items.
  views
    assignment
      assignmentForm.php  // View for form for assigning items to users.
      user.php           // View for displaying a specific user's assignments.
      view.php           // View for displaying user assignments.
    authItem
      AuthChildForm.php  // View for the form for adding children to auth items.
      AuthItemForm.php  // View for form for creating/updating auth items.
      create.php         // View for creating auth items.
      update.php         // View for updating auth items.
    main
      _menu.php          // Partial view for the menu.
      _permissions.php   // Partial view for the permissions. (needed for AJAX)
      operations.php     // View for displaying operations.
      permissions.php    // View for displaying permission overview.
      roles.php          // View for displaying roles.
      tasks.php          // View for displaying tasks.
  RightsModule.php       // Rights module.
```

Change Log

Release 0.9.5

- Support for custom style sheet
- General code improvements

Release 0.9.3

- Installer
- Improved module configuration
- Pagination for Assignments

Release 0.9.2

- Support for I18N (internationalization)

Release 0.9.1

- Rights initial release.

Pre 0.9.1

- Rights was originally developed as Yii-Authorization, but the project name was changed because the module was basically rewritten to improve the logic.

Glossary

Term	Description
ACAC	<i>Stands for Automatic Controller Access Control. Functionality for checking access automatically before running a controller action.</i>
Authorization Item	<i>A permission to do something. Authorization Items can be classified as Operations, Tasks and Roles.</i>
Authorization Manager	<i>A component which manages the Authorization items.</i>
Authorizer	<i>Rights' Authorizer component that provides the module's core functionality.</i>
Assignment	<i>An Authorization Item assigned to a user.</i>
CDbAuthManager	<i>Yii's built-in database Authorization Manager.</i>
Default roles	<i>Roles assigned to all users implicitly.</i>
Filter	<i>Component that is executed before running a controller action.</i>
I18N	<i>See Internationalization.</i>
Installer	<i>Rights' Installer component that allows for easier installation of the module.</i>
Internationalization	<i>Support for different languages and regions.</i>
Operation	<i>See Authorization Item.</i>
Permission	<i>Authorization granted to do something.</i>
Role	<i>See Authorization Item.</i>
Super user	<i>A user with access to everything.</i>
Super user role	<i>Name of the role with super user privileges.</i>
Task	<i>See Authorization Item.</i>