
Yii2-user Documentation

Release 0.6

Dektrium project

December 08, 2014

1	Getting started	3
1.1	Overview	3
1.2	Installation	3
2	Basics	5
2.1	Configuration	5
2.2	List of available actions	6
2.3	User management	6
2.4	Overriding models	7
2.5	Overriding views	9
2.6	Overriding controllers	10
2.7	Social auth	11
3	How-to	13
3.1	Adding captcha to forms	13
4	Feedback	17

Yii2-user is a flexible user registration and authentication module for Yii2.

Getting started

1.1 Overview

Most of web applications provide a way for users to register, log in or reset their forgotten passwords. Rather than re-implementing this on each application, you can use Yii2-user which is a flexible user management module for Yii2 that handles common tasks such as registration, authentication and password retrieval. Current version includes following features:

- Registration, with an optional confirmation per mail
- Social auth
- Password retrieval
- Account and profile management
- Console commands
- User management interface

Note: Module is in initial development. Anything may change at any time.

[Codeception](#) is used for testing and [Travis CI](#) is used for continuous integration.

Yii2-user is an open-source project and released under the [MIT License](#). You are very welcome to contribute to the module.

Yii2-user is a part of **Dektrium Project** which has been started with aim to provide useful extensions to your Yii2 application under one vendor namespace. We hope that by providing extensions to common requirements, we will help developers concentrate on the interesting things that their applications do.

If you have any questions or comments, please use [Google Group](#) or [Gitter](#). You can also get news and updates from the [Blog](#).

1.2 Installation

This document will guide you through the process of installing Yii2-user using **composer**. Installation is a quick and easy two-step process. Installation is fully automatic: you don't even need to configure module manually!

Note: Before we start make sure that you have properly configured **db** and **mail** application components.

1.2.1 Downloading Yii2-user using composer

Add Yii2-user to the require section of your **composer.json** file:

```
{
    "require": {
        "dektrium/yii2-user": "*"
    }
}
```

And run following command to make **composer** download and install Yii2-user:

```
$ php composer.phar update
```

1.2.2 Updating database schema

After you downloaded Yii2-user, the last thing you need to do is update your database schema by applying the migrations:

```
$ php yii migrate/up --migrationPath=@vendor/dektrium/yii2-user/migrations
```

That's all! You have successfully finished Yii2-user installation and from now you are ready to use all its functionality.

2.1 Configuration

This guide covers the basic configuration settings for the Yii2-user.

2.1.1 Available configuration options

- **enableRegistration** Whether registration is enabled. Defaults to **True**.
- **enableGeneratingPassword** Whether password field is not shown on signup page and password is generated automatically and sent to user via email. Defaults to **False**.
- **enableConfirmation** Whether users have to confirm their accounts by clicking confirmation link sent them by email. In order to enable this option you have to configure **mail** application component. Defaults to **True**.
- **enableUnconfirmedLogin** Whether users are allowed to sign in without activating their accounts. Default to **False**.
- **enablePasswordRecovery** Whether users are allowed to recover their passwords. Defaults to **True**.
- **enableEmailReconfirmation** Whether users have to reconfirm their email after changing it on settings page. Defaults to **False**.
- **confirmWithin** The time in seconds before a confirmation token becomes invalid. After expiring this time user have to request new confirmation token on special page. Defaults to **86400** (24 hours).
- **rememberFor** The time in seconds you want the user will be remembered without asking for credentials. Defaults to **1209600** (2 weeks).
- **recoverWithin** The time in seconds before a recovery token becomes invalid. After expiring this time user have to request new recovery message. Defaults to **21600** (6 hours).
- **admins** An array of user's usernames who can manage users from admin panel. Defaults to empty array.
- **cost** Cost parameter used by the Blowfish hash algorithm. Defaults to **10**.
- **urlPrefix** The prefix for user module URL. Defaults to **"user"**.
- **urlRules** The rules to be used in URL management.

2.1.2 Configuration example

The configuration is done in the application's `config/web.php` file.

```
<?php return [
    ...
    'modules' => [
        ...
        'user' => [
            'class' => 'dektrium\user\Module',
            'enableUnconfirmedLogin' => true,
            'confirmWithin' => 21600,
            'cost' => 12,
            'admins' => ['admin']
        ],
        ...
    ],
    ...
];
```

2.2 List of available actions

Yii2-user includes a lot of actions, which you can access by creating URLs for them. Here is the table of available actions which contains route and short description of each action. You can create URLs for them using special Yii helper `\yii\helpers\Url::to()`.

Route	Description
/user/registration/register	Displays registration form
/user/registration/resend	Displays resend form
/user/registration/confirm	Confirms a user (requires <i>id</i> and <i>token</i> query params)
/user/security/login	Displays login form
/user/security/logout	Logs the user out (available only via POST method)
/user/recovery/request	Displays recovery request form
/user/recovery/reset	Displays password reset form (requires <i>id</i> and <i>token</i> query params)
/user/settings/profile	Displays profile settings form
/user/settings/email	Displays email settings form
/user/settings/password	Displays password settings form
/user/profile/show	Displays user's profile (requires <i>id</i> query param)
/user/admin/index	Displays user management interface

2.3 User management

When you start using Yii2-user you will probably find that you need to create, update and delete accounts of other users. In order to do that Yii2-user provides beautiful CRUD interface.

To start using user management interface you have to add your username to administrator's list as follows:

```
<?php return [
    ...
    'modules' => [
        ...
        'user' => [
            'class' => 'dektrium\user\Module',
            'admins' => ['your-username-goes-here']
        ],
        ...
    ],
    ...
];
```

```
    ...  
];
```

2.3.1 Show users

Route `/user/admin/index` shows a list of registered users. You will be able to see a lot of useful information such as registration time and ip address, confirmation and block status, etc.

2.3.2 Create user

Route `/user/admin/create` shows create user form. To create a new user account you have to fill username and email fields. If you do not want to create password for user leave password field empty, password will be generated automatically. After create a welcome message will be sent to email that you have used to create user. It will contain username and password.

2.3.3 Update user

Route `/user/admin/update` shows update user form. To access this route you should specify id query parameter.

2.3.4 Delete user

Route `/user/admin/delete` deletes an user account. To access this route you should specify id query parameter and do a POST request. Be careful, you will not be able to restore deleted account.

2.4 Overriding models

When you are creating application with Yii2-user you can find that you need to override models or forms. Yii2-user is very extensible and allows you to override any model. Yii2-user does not create models using “new” statement, instead it uses component named “ModelManager” which creates requested models. Here is default model manager configuration:

```
<?php  
return [  
    ...  
    'modules' => [  
        ...  
        'user' => [  
            'class' => 'dektrium\user\Module',  
            'components' => [  
                'manager' => [  
                    // Active record classes  
                    'userClass' => 'dektrium\user\models\User',  
                    'tokenClass' => 'dektrium\user\models\Token',  
                    'profileClass' => 'dektrium\user\models\Profile',  
                    'accountClass' => 'dektrium\user\models\Account',  
                    // Model that is used on user search on admin pages  
                    'userSearchClass' => 'dektrium\user\models\UserSearch',  
                    // Model that is used on registration  
                    'registrationFormClass' => 'dektrium\user\models\RegistrationForm',  
                    // Model that is used on resending confirmation messages
```

```
        'resetFormClass' => 'dektrium\user\models\ResetForm',
        // Model that is used on logging in
        'loginFormClass' => 'dektrium\user\models\LoginForm',
        // Model that is used on password recovery
        'passwordRecoveryFormClass' => 'dektrium\user\models\RecoveryForm',
        // Model that is used on requesting password recovery
        'passwordRecoveryRequestFormClass' => 'dektrium\user\models\RecoveryRequestForm',
    ],
    ],
    ...
],
...
];
```

2.4.1 Example

Assume you decided to override user class and change registration process. Let's create new user class under `@app/models`.

```
<?php
namespace app\models;

use dektrium\user\models\User as BaseUser;

class User extends BaseUser
{
    public function register()
    {
        // do your magic
    }
}
```

In order to make Yii2-user use your class you need to configure manager component as follows:

```
<?php
return [
    ...
    'modules' => [
        ...
        'user' => [
            'class' => 'dektrium\user\Module',
            'components' => [
                'manager' => [
                    'userClass' => 'app\models\User',
                ],
            ],
        ],
    ],
];
```

Well done! Yii2-user now uses your User model.

2.5 Overriding views

When you start using Yii2-user you will probably find that you need to override the default views provided by the module. Although view names are not configurable, Yii2 provides a way to override views using themes. To get started configure your view application component as follows:

```
<?php return [
    ...
    'components' => [
        'view' => [
            'theme' => [
                'pathMap' => [
                    '@dektrium/user/views' => '@app/views/user'
                ],
            ],
        ],
    ],
    ...
];
```

In the above **pathMap** means that every view in `@dektrium/user/views` will be first searched under `@app/views/user` and if a view exists in the theme directory it will be used instead of the original view.

2.5.1 Example

An example of overriding the registration page view is demonstrated below. First make sure you have configured view application component.

In order to override the registration view file you have to create `@app/views/user/registration/register.php`. Open it and paste in the following code:

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;

/**
 * @var yii\web\View $this
 * @var yii\widgets\ActiveForm $form
 * @var dektrium\user\models\User $user
 */

$this->title = Yii::t('user', 'Sign up');
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="alert alert-success">
    <p>This view file has been overridden!</p>
</div>
<div class="row">
    <div class="col-md-4 col-md-offset-4">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title"><?= Html::encode($this->title) ?></h3>
            </div>
            <div class="panel-body">
                <?php $form = ActiveForm::begin([
                    'id' => 'registration-form',
                ]); ?>
```

```
<?= $form->field($model, 'username') ?>

<?= $form->field($model, 'email') ?>

<?= $form->field($model, 'password')->passwordInput() ?>

<?= Html::submitButton(Yii::t('user', 'Sign up'), ['class' => 'btn btn-success btn-b

    <?php ActiveForm::end(); ?>
</div>
</div>
<p class="text-center">
    <?= Html::a(Yii::t('user', 'Already registered? Sign in!'), ['/user/security/login']) ?>
</p>
</div>
</div>
```

Then open registration page and make sure that you see **‘This view file has been overridden!’**. If you don’t make sure you have properly configured your view component and created view file in needed location.

2.6 Overriding controllers

Sometimes you may need to override default Yii2-user controllers. It is pretty easy and takes two steps.

2.6.1 Step 1: Create new controller

First of all you need to create new controller under your own namespace (we’d recommend `app\controllers\user`) and extend it from needed Yii2-user controller.

For example, if you want to override `AdminController` you should create `app\controllers\user\AdminController` and extend it from `dektrium\user\controllers\AdminController`:

```
<?php

namespace app\controllers\user;

use dektrium\user\controllers\AdminController as BaseAdminController;

class AdminController extends BaseAdminController
{
    public function actionCreate()
    {
        // do your magic
    }
}
```

2.6.2 Step 2: Add your controller to controller map

To let Yii2-user know about your controller you should add it to controller map as follows:

```
<?php return [
    ...
    'modules' => [
```

```

...
'admin' => 'app\controllers\user\AdminController'
],
...
],
...
],

```

2.7 Social auth

Yii2-user supports authentication using social networks accounts. It allows to connect social accounts to user account and use them to log in.

2.7.1 Getting started

To get started you need to setup auth client collection application component:

```

<?php
return [
    ...
    'components' => [
        ...
        'authClientCollection' => [
            'class' => 'yii\authclient\Collection',
            'clients' => [
                'google' => [
                    'class' => 'yii\authclient\clients\GoogleOpenId'
                ],
                'facebook' => [
                    'class' => 'yii\authclient\clients\Facebook',
                    'clientId' => 'facebook_client_id',
                    'clientSecret' => 'facebook_client_secret',
                ],
            ],
        ],
        ...
    ],
    ...
];

```

2.7.2 How it works

When you are going to log in you can click social network icon. If you have already logged in using that account you will be logged in. Otherwise you will be shown simple sign up form with two field (username and email).

After you logged in you can go to accounts settings page and connect new account or disconnect already connected accounts.

3.1 Adding captcha to forms

Adding captcha to forms is pretty easy and can be done in three steps:

1. In the model you have to add captcha field and validation rules.
2. In the view you have to show captcha field
3. In the controller you have to add captcha action

In this howto I would like to show you how to add captcha field in the registration form but you can add captcha to any form following this steps.

3.1.1 1. Adding field and validation rules to model

First of all you need to override Registration form as described in special guide. After this done you have to add public property named **captcha** and validation rules.

```
<?php

namespace app\models;

class RegistrationForm extends \dektrium\user\models\RegistrationForm
{
    /**
     * @var string
     */
    public $captcha;
    /**
     * @inheritdoc
     */
    public function rules()
    {
        $rules = parent::rules();
        $rules[] = ['captcha', 'required'];
        $rules[] = ['captcha', 'captcha'];
        return $rules;
    }
}
```

3.1.2 2. Adding widget to the view

Before doing this step you have to configure view application component as described in guide. After this done you have to create new file named **register.php** in **@app/views/user/registration**. Now you have to add widget to registration form, just copy and paste following code into newly created view file.

```
<?php

use yii\helpers\Html;
use yii\widgets\ActiveForm;
use yii\captcha\Captcha;

/**
 * @var yii\web\View $this
 * @var yii\widgets\ActiveForm $form
 * @var app\models\RegistrationForm $model
 */
$this->title = Yii::t('user', 'Sign up');
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="row">
    <div class="col-md-4 col-md-offset-4">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title"><?= Html::encode($this->title) ?></h3>
            </div>
            <div class="panel-body">
                <?php $form = ActiveForm::begin([
                    'id' => 'registration-form',
                ]); ?>

                <?= $form->field($model, 'username') ?>

                <?= $form->field($model, 'email') ?>

                <?= $form->field($model, 'password')->passwordInput() ?>

                <?= $form->field($model, 'captcha')->widget(Captcha::className()) ?>

                <?= Html::submitButton(Yii::t('user', 'Sign up'), ['class' => 'btn btn-success btn-b...

                <?php ActiveForm::end(); ?>
            </div>
        </div>
        <p class="text-center">
            <?= Html::a(Yii::t('user', 'Already registered? Sign in!'), ['/user/security/login']) ?>
        </p>
    </div>
</div>
```

3.1.3 3. Adding action to the controller

In order to make captcha work you have to add captcha action to **app\controllers\SiteController** Maybe it is already added because standard Yii2 application template adds it automatically.

```
<?php
namespace app\controllers;
```

```
class SiteController extends \yii\web\Controller
{
    ...
    public function actions()
    {
        return [
            'captcha' => [
                'class' => 'yii\captcha\CaptchaAction',
            ],
        ];
    }
    ...
}
```

Feedback

You are very welcome to help us improve the quality of Yii2-user documentation. Please contribute if you see any mistakes or actual errors. To get started you can read the [Contributing guide](#).

If you can not fix the bug or mistake by yourself, please open an [issue](#).

Any kind of discussion about Yii2-user documentation is very welcome in our [Google group](#) or [Gitter](#).