

 무료 전자 책

배우기

yii2

Free unaffiliated eBook created from
Stack Overflow contributors.

#yii2

.....	1
1: yii2	2
.....	2
.....	2
Examples.....	2
.....	2
Composer	2
.....	2
Yii	3
.....	3
Yii2	3
2: Ajax	6
Examples.....	6
Ajax	6
Ajax.....	7
3: Pjax	10
Examples.....	10
1	10
2	10
pjax	10
pjax	10
pjax	11
Pjax	11
4: Yii2 ActiveForm	13
Examples.....	13
Yii2	13
ActiveForm	14
5: Yii2 OAuth2 - : Facebook OAuth2	15
Examples.....	15
.....	15
yii2-authclient	16

.....	17
Facebook redirect_url	18
onAuthSuccess	18
6:	20
Examples.....	20
.....	20
.....	20
.....	20
symlink	21
7:	22
Examples.....	22
.....	22
.....	22
8:	24
Examples.....	24
.....	24
.....	24
.....	24
.....	24
.....	25
/ /	25
.....	25
.....	25
.....	25
.....	26
.....	26
.....	26
.....	26
.....	26
9:	27
Examples.....	27
Yii2	27

where ()	28
orderBy ()	29
10: URL	31
.....	31
Examples	31
URL	31
11:	33
.....	33
Examples	33
.....	33
12:	34
Examples	34
yii2	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	34
.....	35
.....	35
URL	35
13:	37
Examples	37
Composer Extension	37
14: API	38
Examples	38
API	38
API Yii2	39

Content-Type	40
15:	41
.....	41
.....	41
Examples.....	41
.....	41
.....	41
HTML.....	42
16:	44
.....	44
Examples.....	44
.....	44
.....	44
.....	44
ID	44
.....	45
17:	46
Examples.....	46
.....	46
ActiveRecord	46
18: Yii2 JQuery	48
Examples.....	48
max jquery	48
jquery	48
jquery ().....	48
19:	50
Examples.....	50
.....	50
.....	50
.....	50
.....	50
.....	51

.....	51
20:	54
Examples.....	54
Yii2	54
:	55
/	55
.....	56
.....	57
21:	58
.....	58
Examples.....	58
.....	58
Where	58
ActiveRecord	60
.....	61
.....	61
.....	61
.....	63

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [yii2](#)

It is an unofficial and free yii2 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official yii2.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: yii2

Yii PHP . , , CMS (Content Management System), , RESTful .

2.0.12	2017-06-05
2.0.11	2017-02-01
2.0.10	2016-10-20
2.0.9	2016-07-11
2.0.8	2016-04-28
2.0.7	2016-02-14
2.0.6	2015-08-06
2.0.5	2015-07-11
2.0.4	2015-05-10
2.0.3	2015-03-01
2.0.2	2015-01-11
2.0.1	2014-12-07
2.0.0	2014-10-12

Examples

Yii2 .

1. Composer
- 2.

Composer

Composer getcomposer.org . Linux Mac OS X .

```
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer
```

Windows [composer-setup.exe](#) . Github API [github API](#) .

Yii

Composer Yii .

```
composer global require "fxp/composer-asset-plugin:^1.2.0"
composer create-project --prefer-dist yiisoft/yii2-app-basic basic
```

Yii2 .

```
composer create-project --prefer-dist --stability=dev yiisoft/yii2-app-basic basic
```

Yii2

```
composer create-project --prefer-dist --stability=dev yiisoft/yii2-app-advanced advanced
cd advanced
php init
```

common / config / main-local.php ['db'] .

```
php yii migrate
```

1. [Yii- archive](#)

2. .

3. `cookieValidationKey` config / web.php

```
'cookieValidationKey' => '',
```

For example : xyctuyvibonp

```
'cookieValidationKey' => 'xyctuyvibonp',
```

```
//insert a secret key in the following (if it is empty) - this is required by cookie
validation
```

```
'cookieValidationKey' => 'enter your secret key here',
```

Yii2 .

```
curl -sS https://getcomposer.org/installer | php
```

```
sudo mv composer.phar /usr/local/bin/composer
```

```
composer
```

Composer .

Yii2 .

1.Installing

zip .

(: /var/www/html .

<https://github.com/yiisoft/yii2>

"" . .

```
cd advanced
```

.

```
php init
```

2.

github . GitHub .

.

1..

2..

3..

4..

5..

6..

7.. .

: <https://help.github.com/articles/creating-an-access-token-for-command-line-use/>

```
cd /var/www/html/
```

```
composer config -g github-oauth.github.com <AuthToken>
```

:

```
composer config -g github-oauth.github.com fleefb8f188c22dd6467f1883cb2615c194d1ce1
```

yii2

```
composer create-project --prefer-dist yiisoft/yii2-app-advanced advanced
```

```
"" . .
```

```
cd advanced
```

```
.
```

```
php init
```

```
!
```

```
.
```

[http : // localhost / advanced / frontend / web](http://localhost/advanced/frontend/web)

[http : // localhost / advanced / backend / web](http://localhost/advanced/backend/web)

[yii2](https://riptutorial.com/ko/yii2/topic/788/yii2-) : <https://riptutorial.com/ko/yii2/topic/788/yii2->

2: Ajax

Examples

Ajax

:

```
<?php
use yii;
use yii\bootstrap\ActiveForm;
use yii\helpers\Html;
?>

<?php
$form = ActiveForm::begin([
    'action' => ['comments/ajax-comment'],
    'options' => [
        'class' => 'comment-form'
    ]
]);
?>

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

<?= Html::submitButton("Submit", ['class' => "btn"]); ?>

<?php ActiveForm::end(); ?>
```

:

```
jQuery(document).ready(function($) {
    $(".comment-form").submit(function(event) {
        event.preventDefault(); // stopping submitting
        var data = $(this).serializeArray();
        var url = $(this).attr('action');
        $.ajax({
            url: url,
            type: 'post',
            dataType: 'json',
            data: data
        })
        .done(function(response) {
            if (response.data.success == true) {
                alert("Wow you commented");
            }
        })
        .fail(function() {
            console.log("error");
        });
    });
});
```

:

```

public function actionAjaxComment()
{
    $model = new Comments();
    if (Yii::$app->request->isAjax) {
        Yii::$app->response->format = \yii\web\Response::FORMAT_JSON;

        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return [
                'data' => [
                    'success' => true,
                    'model' => $model,
                    'message' => 'Model has been saved.',
                ],
                'code' => 0,
            ];
        } else {
            return [
                'data' => [
                    'success' => false,
                    'model' => null,
                    'message' => 'An error occurred.',
                ],
                'code' => 1, // Some semantic codes that you know them for yourself
            ];
        }
    }
}

```

Ajax

Controller::renderAjax() Ajax . renderPartial() JS / CSS .

```

<?php
use yii\helpers\Html;
use yii\bootstrap\ActiveForm;

\yii\bootstrap\BootstrapAsset::register($this);

<div class="site-login">

    <?php $form = ActiveForm::begin(); ?>

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

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

    <?= Html::submitButton('Login', ['class' => 'btn btn-primary btn-block']) ?>

    <?php ActiveForm::end(); ?>
</div>

```

renderPartial() :

```

public function actionLogin()
{

```

```

$model = new LoginForm();
if ($model->load(Yii::$app->request->post()) && $model->login()) {
    return $this->goBack();
}
return $this->renderPartial('login', [
    'model' => $model,
]);
}

```

:

```

<div class="site-login">
    <form id="w0" action="/site/login" method="post" role="form">
        <div class="form-group field-loginform-username required">
            <label class="control-label" for="loginform-username">Имя пользователя</label>
            <input type="text" id="loginform-username" class="form-control"
name="LoginForm[username]">
        </div>
        <div class="form-group field-loginform-password required">
            <label class="control-label" for="loginform-password">Пароль</label>
            <input type="password" id="loginform-password" class="form-control"
name="LoginForm[password]">
        </div>
        <button type="submit" class="btn btn-primary btn-block">Login</button>
    </form>
</div>

```

renderAjax() :

```

...
return $this->renderAjax('login', [
    'model' => $model,
]);
...

```

(JS, CSS) :

```

<link href="/assets/f1759119/css/bootstrap.css" rel="stylesheet">
<div class="site-login">
    <form id="w0" action="/site/login" method="post" role="form">
        <div class="form-group field-loginform-username required">
            <label class="control-label" for="loginform-username">Имя пользователя</label>
            <input type="text" id="loginform-username" class="form-control"
name="LoginForm[username]">
        </div>
        <div class="form-group field-loginform-password required">
            <label class="control-label" for="loginform-password">Пароль</label>
            <input type="password" id="loginform-password" class="form-control"
name="LoginForm[password]">
        </div>
        <button type="submit" class="btn btn-primary btn-block">Login</button>
    </form>
</div>
<script src="/assets/13aa7b5d/jquery.js"></script>
<script src="/assets/302a2946/yii.js"></script>
<script src="/assets/302a2946/yii.validation.js"></script>
<script src="/assets/302a2946/yii.activeForm.js"></script>

```

():

```
...
Yii::$app->assetManager->bundles = [
    'yii\bootstrap\BootstrapAsset' => false,
];
return $this->renderAjax('login', [
    'model' => $model,
]);
...
```

(no bootstrap.css) :

```
<div class="site-login">
  <form id="w0" action="/site/login" method="post" role="form">
    <div class="form-group field-loginform-username required">
      <label class="control-label" for="loginform-username">Имя пользователя</label>
      <input type="text" id="loginform-username" class="form-control"
name="LoginForm[username]">
    </div>
    <div class="form-group field-loginform-password required">
      <label class="control-label" for="loginform-password">Пароль</label>
      <input type="password" id="loginform-password" class="form-control"
name="LoginForm[password]">
    </div>
    <button type="submit" class="btn btn-primary btn-block">Login</button>
  </form>
</div>
<script src="/assets/13aa7b5d/jquery.js"></script>
<script src="/assets/302a2946/yii.js"></script>
<script src="/assets/302a2946/yii.validation.js"></script>
<script src="/assets/302a2946/yii.activeForm.js"></script>
```

Ajax : <https://riptutorial.com/ko/yii2/topic/2944/ajax->

3: Pjax

Examples

1

views \ site \ form-submission.php

```
<?php Pjax::begin(['id'=>'id-pjax']); ?>
<?= Html::beginForm(['site/form-submission'], 'post', ['data-pjax' => '', 'class' => 'form-inline']); ?>
    <?= Html::input('text', 'string', Yii::$app->request->post('string'), ['class' => 'form-control']); ?>
    <?= Html::submitButton('Hash String', ['class' => 'btn btn-lg btn-primary', 'name' => 'hash-button']); ?>
<?= Html::endForm(); ?>
<h3><?= $stringHash ?></h3>
<?php Pjax::end(); ?>
```

2

```
public function actionFormSubmission()
{
    $security = new Security();
    $string = Yii::$app->request->post('string');
    $stringHash = '';
    if (!is_null($string)) {
        $stringHash = $security->generatePasswordHash($string);
    }
    return $this->render('form-submission', [
        'stringHash' => $stringHash,
    ]);
}
```

pjax

.

```
<?php
use yii\widgets\Pjax;
?>
```

.

```
<?php Pjax::begin(['id'=>'id-pjax']); ?>
Content that needs to be updated
<?php Pjax::end(); ?>
```

pjax


```
$.pjax.reload({container: '#id-pjax'});
```

pjax

```
<?php Pjax::begin(['id'=>'id-pjax', 'timeout' => false]); ?>
```

timeout . (1000).

pjax GET . POST .

```
<?php Pjax::begin(['id'=>'id-pjax', 'timeout' => false, 'clientOptions' => ['method' => 'POST']]); ?>
```

Pjax

Yii Framework 2.0 JavaScript [Pjax](#) . Ajax .

: 1 . 2 ActiveForm . ActiveForm JavaScript JavaScript , Page 1 activeform
JavaScript . 'Uncaught TypeError : undefined is not '.

: Active .

```
class AppAsset extends AssetBundle
{
    ...
    public $depends = [
        'yii\widgets\ActiveFormAsset',
        'yii\validators\ValidationAsset',
    ];
    ...
}
```

: 1 (NavBar) . 2 (ActiveForm) . Pjax JavaScript ActiveForm . ActiveForm
init 'this' ActiveForm . NavBar div . div ID , ActiveForm # w1 ID NavBar NavBar ID .

: ID Yii ., ID ID .

: Pjax 1,000ms .

: Pjax . 1 . xdebug .

: [PRG \(Post-Redirect-Get\)](#) . Pjax .

: Pjax . Pjax , Pjax . . .

```
$sendURL = "main/endpoint";
if (Yii::$app->request->isPjax) {
    return $this->run($sendURL);
} else {
    return $this->redirect([$sendURL]);
}
```

Pjax Yii ? !

Pjax : <https://riptutorial.com/ko/yii2/topic/4554/pjax>

4: Yii2 ActiveForm

Examples

Yii2

Yii2

yii2 ActiveForm .

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

id form .

```
$form =ActiveForm::begin([ 'id'=> 'login-form', 'options'=> ['class' => 'form-  
horizontal'],]) ?>
```

\$ model .

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

'username' 'password' .

CSS " .

```
<div class="form-group">  
  <div class="col-lg-offset-1 col-lg-11">  
    <?= Html::submitButton('Login', ['class' => 'btn btn-primary']) ?>  
  </div>  
</div>
```

.

```
<?php ActiveForm::end() ?>
```

:

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

TextField :

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

:

```
echo $form->field($model, 'hidden1')->hiddenInput()->label(false);
```

:

```
<?php echo $form->field($model, 'name')
->dropDownList (
Stud::find()->select(['name'])
->indexBy('name')->column(),
['prompt'=>'Select no']); ?>
```

Id

```
<?= $form->field($model, 'name')->dropDownList (
    ArrayHelper::map(Stud::find()->all(), 'no', 'name'), ['prompt' => 'Select Car
Name']
) ?>
```

FileUploader :

```
echo $form->field($model, 'imagepath')->fileInput();
```

```
<?= $form->field($model, 'username')->textInput()->hint('Please enter your name')-
>label('Name') ?>
```

ActiveForm

/ .

```
$form = ActiveForm::begin([
    'id' => 'signup-form',
    'enableClientValidation' => true,
    'enableAjaxValidation' => true,
    'validationUrl' => Url::to('signup'),
]);
```

1. enableClientValidation ActiveForm . false .
2. enableAjaxValidation ActiveForm . ActiveForm .
3. validationUrl - ActiveForm validationUrl . .

. ajax enableAjaxValidation . .

. enableAjaxValidation .

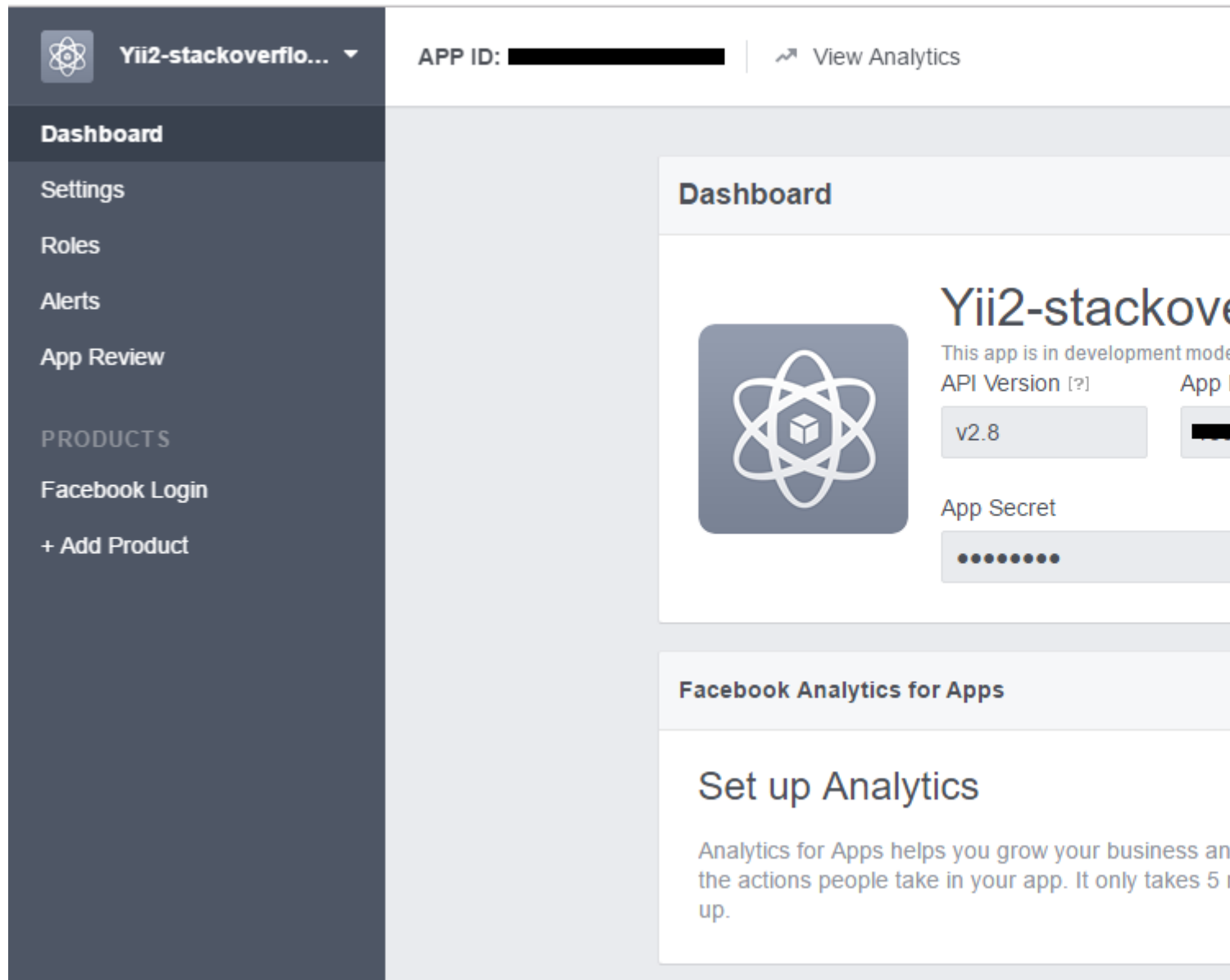
```
echo $form->field($model, 'username', ['enableAjaxValidation' => true]);
```

Yii2 ActiveForm : <https://riptutorial.com/ko/yii2/topic/6807/yii2-activeform>

5: Yii2 OAuth2 - : Facebook OAuth2

Examples

<https://developers.facebook.com/> .



Add product Facebook Login .

Client OAuth Settings

 Yes

Client OAuth Login

Enables the standard OAuth client token flow. Secure your application and prevent abuse by specifying which token redirect URIs are allowed with the options below. Disable globally if not used.

 Yes

Web OAuth Login

Enables web based OAuth client login for building custom login flows. [?]

 No

Force Web OAuth Login

When on, prompts users to use web based login. [?]

 No

Embedded Browser OAuth Login

Enables browser control redirect uri for OAuth client login. [?]

Valid OAuth redirect URIs

 No

Login from Devices

Enables the OAuth client login flow for devices like a smart TV [?]

Deauthorize

Deauthorise Callback URL

yii2-authclient

yii2-app . yii2-basic . .

```
composer require --prefer-dist yiisoft/yii2-authclient
```

```
"yiisoft/yii2-authclient": "~2.1.0"
```

```
composer.json require composer.json .
```

```
config components config authClientCollection .
```

```
return [
    'components' => [
        'authClientCollection' => [
            'class' => 'yii\authclient\Collection',
        ],
    ],
];
```

```

        'clients' => [
            'facebook' => [
                'class' => 'yii\authclient\clients\Facebook',
                'clientId' => 'facebook_client_id',
                'clientSecret' => 'facebook_client_secret',
            ],
        ],
    ],
    // ...
];

```

facebook_client_id ID facebook_client_secret .

1. Login as facebook account Login as facebook account :

site/login.php login :

```

<?= yii\authclient\widgets\AuthChoice::widget([
    'baseAuthUrl' => ['site/auth'],
    'popupMode' => false,
]) ?>

```

SiteController auth OAuth2 SiteController .

```

class SiteController extends Controller
{
    public function actions()
    {
        return [
            'auth' => [
                'class' => 'yii\authclient\AuthAction',
                'successCallback' => [$this, 'onAuthSuccess'],
            ],
        ];
    }

    public function onAuthSuccess($client)
    {
        // do many stuff here, save user info to your app database
    }
}

```

yii\authclient\AuthAction URL .

onAuthSuccess .

Facebook redirect_url

yii2-app prettyUrl redirect_uri .

```
http://<base_url>/web/site/auth
```

URL .

```
http://<base_url>/web/index.php?r=site%2Fauth
```

:

Client OAuth Settings

Yes **No** **Client OAuth Login**
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

Yes **No** **Web OAuth Login**
Enables web based OAuth client login for building custom login flows. [?]

No **Force Web OAuth Reauthentication**
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

No **Embedded Browser OAuth Login**
Enables browser control redirect uri for OAuth client login. [?]

Valid OAuth redirect URIs

|

No **Login from Devices**
Enables the OAuth client login flow for devices like a smart TV [?]

onAuthSuccess

```
/**  
 * @param $client ClientInterface  
 */  
public function onAuthSuccess($client)  
{  
    //Get user info  
    /** @var array $attributes */
```



```
$attributes = $client->getUserAttributes();
$email = ArrayHelper::getValue($attributes, 'email'); //email info
$id = ArrayHelper::getValue($attributes, 'id'); // id facebook user
$name = ArrayHelper::getValue($attributes, 'name'); // name facebook account

//Login user
//For demo, I will login with admin/admin default account
$admin = User::findByUsername('admin');
Yii::$app->user->login($admin);
}
```

Yii2 OAuth2 - : Facebook OAuth2 : <https://riptutorial.com/ko/yii2/topic/7428/yii2-oauth2-----facebook-oauth2>

6:

Examples

URL <http://site.local> URL <http://site.local/admin>

webroot (: www public_html). www

```
www
  admin
  backend
  common
  console
  environments
  frontend
  ...
```

www / . / www / admin . index.php index-test.php .

/ config / main.php .

```
'components' => [
    'request' => [
        'csrfParam' => '_csrf-backend',
        'csrfCookie' => [
            'httpOnly' => true,
            'path' => '/admin',
        ],
    ],
],
'user' => [
    'identityClass' => 'common\models\User',
    'enableAutoLogin' => true,
    'identityCookie' => [
        'name' => '_identity-backend',
        'path' => '/admin',
        'httpOnly' => true,
    ],
],
'session' => [
    // this is the name of the session cookie used for login on the backend
    'name' => 'advanced-backend',
    'cookieParams' => [
        'path' => '/admin',
    ],
],
],
```

: <https://github.com/yiisoft/yii2-app-advanced/blob/master/docs/guide/topic-shared-hosting.md>

symlink

say /backend/web/uploads/ /backend/web/uploads/ . .

```
ln -s /path/to/backend/web/uploads/ /path/to/frontend/web/uploads
```

```
<img src='/uploads/<?= $model->image?>' alt='My Image goes here'>  
<a href='/uploads/<?= $model->filename?>' target='_blank'>Download File</a>
```

: <https://riptutorial.com/ko/yii2/topic/944/-->

7:

Examples

:

- components .
- : MyComponent.php

```
namespace app\components;

use Yii;
use yii\base\Component;
use yii\base\InvalidConfigException;

class MyComponent extends Component
{
    public function demo()
    {
        return "welcome";
    }
}
```

- config/web.php

```
components' => [
    'mycomponent' => [
        'class' => 'app\components\MyComponent',
    ],
]
```

.

```
namespace app\controllers;

use Yii;

class DemoController extends \yii\web\Controller
{
    public function actionTest()
    {
        echo Yii::$app->mycomponent->demo();
    }
}
```

MyComponent.php

```
namespace app\components;

use Yii;
use yii\base\Component;
```

```

use yii\base\InvalidConfigException;
use yii\helpers\Url;
use yii\helpers\ArrayHelper;

use app\models\User;

class MyComponent extends Component
{
    // Function return list of id & user Names,used for dropdownlist
    public function getUserID()
    {
        $code = User::find()->select('id,name')
        ->where(['is_deleted'=>'n'])
        ->all();

        $result = ArrayHelper::map($code, 'id', 'name');
        if($result)
            return $result;
        else
            return ["null"=>"No User"];
    }
}

```

-> web.php

```

components' => [
    'mycomponent' => [
        'class' => 'app\components\MyComponent',
    ],
]

```

-> .

```

<?= $form->field($model, 'user_id')->dropDownList(Yii::$app->mycomponent->getUserID())?>

```

: <https://riptutorial.com/ko/yii2/topic/2217/>

8:

Examples

```
yii migrate/create <name>
```

```
. news create_news_table
```

```
yii migrate/create create_news_table
```

```
<?php  
  
use yii\db\Migration;  
  
class m150101_185401_create_news_table extends Migration  
{  
    public function up()  
    {  
  
    }  
  
    public function down()  
    {  
        echo "m101129_185401_create_news_table cannot be reverted.\n";  
  
        return false;  
    }  
  
    /*  
    // Use safeUp/safeDown to run migration code within a transaction  
    public function safeUp()  
    {  
    }  
  
    public function safeDown()  
    {  
    }  
    */  
}
```

```
public function up()  
{  
    $this->dropTable('post');  
}
```

```
yii migrate/create create_post_table --fields="title:string,body:text"
```

```
:
```

```

/**
 * Handles the creation for table `post`.
 */
class m150811_220037_create_post_table extends Migration
{
/**
 * @inheritdoc
 */
public function up()
{
    $this->createTable('post', [
        'id' => $this->primaryKey(),
        'title' => $this->string(),
        'body' => $this->text(),
    ]);
}

/**
 * @inheritdoc
 */
public function down()
{
    $this->dropTable('post');
}
}

```

```

public function up()
{
    $this->createTable('post', [
        'id' => $this->primaryKey()
    ]);
}

```

//

```

public function up()
{
    $this->dropColumn('post', 'position');

    $this->renameColumn('post', 'owner_id', 'user_id');

    $this->alterColumn('post', 'updated', $this->timestamp()->notNull()->defaultValue('0000-00-00 00:00:00'));
}

```

```

public function up()
{
    $this->addColumn('post', 'position', $this->integer());
}

```

```

yii migrate/down      # revert the most recently applied migration
yii migrate/down 3    # revert the most 3 recently applied migrations

```

```

public function safeUp()
{

```

```

$this->createTable('news', [
    'id' => $this->primaryKey(),
    'title' => $this->string()->notNull(),
    'content' => $this->text(),
]);

$this->insert('news', [
    'title' => 'test 1',
    'content' => 'content 1',
]);
}

public function safeDown()
{
    $this->delete('news', ['id' => 1]);
    $this->dropTable('news');
}

```

safeUp() safeDown() safeDown() . up() down() ., .

db . db .

```
yii migrate --db=db2
```

```
yii migrate/redo          # redo the last applied migration
yii migrate/redo 3       # redo the last 3 applied migrations
```

```
yii migrate/history      # showing the last 10 applied migrations
yii migrate/history 5    # showing the last 5 applied migrations
yii migrate/history all  # showing all applied migrations
```

```
yii migrate/new          # showing the first 10 new migrations
yii migrate/new 5        # showing the first 5 new migrations
yii migrate/new all      # showing all new migrations
```

```
yii migrate/mark 150101_185401          # using timestamp to specify the migration
yii migrate/mark "2015-01-01 18:54:01" # using a string that can be parsed by
strtotime()
yii migrate/mark m150101_185401_create_news_table # using full name
yii migrate/mark 1392853618            # using UNIX timestamp
```

```
yii migrate
```

. timestamp up () safeUp () . .

```
yii migrate 3
yii migrate/to 150101_185401          # using timestamp to specify the migration
yii migrate/to "2015-01-01 18:54:01" # using a string that can be parsed by
strtotime()
yii migrate/to m150101_185401_create_news_table # using full name
yii migrate/to 1392853618            # using UNIX timestamp
```

: <https://riptutorial.com/ko/yii2/topic/1929/>-

9:

Examples

Yii2

Yii2 `. emp_id, emp_name emp_salary . .`

```
select emp_name,emp_salary from employee
```

Yii2 `. yii\db\Query .`

```
//creates a new \yii\db\Query() object
$query=new \yii\db\Query();

$rows=$query->select(['emp_name','emp_salary']) //specify required columns in an array
    ->from('employee') //specify table name
    ->all(); //returns an array of rows with each row being an associative array of
name-value pairs.
```

foreach `$ rows - .`

```
foreach ($rows as $row) {
    echo "Employee Name: ".$row['emp_name'].",Employee Salary: ".$row['emp_salary']."<br>";
}
```

.

: Kiran, : 25000

: Midhun, : 50000

: Jishnu, : 20000

: Ajith, : 25000

: Akshay, : 750000

25000 `. sql .`

```
select emp_name from employee where salary=25000
```

Yii2

```
$query=new \yii\db\Query();

$rows=$query->select(['emp_name'])
    ->from('employee')
    ->where(['emp_salary'=>25000]) //specify the condition as an associative array
```

```
where key is column name
->all();
```

25000 Yii2 .

```
$rows=$query->select(['emp_name'])
->from('employee')
->where(['>', 'emp_salary', 25000])
//Here first element in the above array specify relational operator used, second element
specify the table name and third the value itself.
->all();
```

where ()

where () .

```
// Creates a new \yii\db\Query() object
$query = new \yii\db\Query();
$rows = $query->select(['emp_name', 'emp_salary'])
->from('employee')
->where(['emp_name' => 'Kiran', 'emp_salary' => 25000]) // Specify multiple conditions
->one(); // Returns the first row of the result
```

kiran salary 25000 . , () . all () .

all () . 1 0 . . one false .

sql .

```
select emp_name, emp_salary from employee where emp_name = 'Kiran' and emp_salary = 25000
limit 1;
```

Yii2 .

```
$rows = $query->select(['emp_name', 'emp_salary'])
->from('employee')
->where(['emp_name' => 'Kiran'])
->andWhere(['emp_salary' => 25000])
->one();
```

andWhere . . .

where () . . .

```
$rows = $query->select(['emp_name', 'emp_salary'])
->from('employee')
->where(['and', 'emp_name="kiran"', 'emp_salary=25000'])
->one();
```

'and' . , '""', ", ", " , 'NOT IN', ", '""', ' ', '>', '<=' .

'in' 'like'

20000, 25000 50000 . SQL .

```
select * from employee where salary in (20000,25000,50000)
```

Yii2 .

```
$rows = $query->from('employee')
    ->where(['emp_salary' => [20000,25000,50000]])
    ->all();
```

```
$rows = $query->from('employee')
    ->where(['in', 'emp_salary', [20000,25000,50000]]) // Making use of operator format of
where() method
    ->all();
```

20000, 25000 50000 ' in ' not in ' in ' .

where () ' like ' . ' gopal ' . venugopal, rajagopal, gopalakrishnan . SQL .

```
select * from employee where emp_name like '%gopal%'
```

Yii2 .

```
$rows = $query->from('employee')
    ->where(['like', 'emp_name', 'gopal']) // Making use of operator format of where()
method
    ->all();
```

' gopal ' ' nair ' .

```
$rows = $query->from('employee')
    ->where(['like', 'emp_name', ['gopal','nair']]) // Making use of operator format of
where() method
    ->all();
```

```
emp_name '% gopal %' '% nair %' () .
```

' gopal ' ' nair ' ' ' ' .

orderBy ()

orderBy () SQL ORDER BY . emp_id, emp_first_name, emp_last_name emp_salary . . sql .

```
Select * from employee order by emp_salary
```

yii2 .

```
//creates a new \yii\db\Query() object
$query=new \yii\db\Query();

$rows= $query->from('employee')->orderBy([
    'emp_salary' => SORT_ASC //specify sort order ASC for ascending DESC for descending
])->all();
```

, SQL .

```
Select * from employee order by emp_first_name ASC, emp_salary DESC
```

SQL yii2 .

```
//creates a new \yii\db\Query() object
$query=new \yii\db\Query();

$rows= $query->from('employee')->orderBy([
    'emp_first_name' => SORT_ASC
    'emp_salary' => SORT_DESC
])->all();
```

SQL ORDER BY .

```
//creates a new \yii\db\Query() object
$query=new \yii\db\Query();
$rows=$query->from('employee')->orderBy('emp_first_name ASC, emp_salary DESC')->all();
```

addOrderBy () ORDER BY .

```
//creates a new \yii\db\Query() object
$query=new \yii\db\Query();
$rows=$query->from('employee')->orderBy('emp_first_name ASC')
->addOrderBy('emp_salary DESC')->all();
```

: <https://riptutorial.com/ko/yii2/topic/4167/>-

10: URL

URL `yii\helpers\Url` . `urlManager url` .

Examples

URL

`\helpers\url` URL . / .

URL:

```
echo Url::to(['post/index']);
```

URL:

```
echo Url::to(['post/view', 'id' => 100]);
```

URL:

```
echo Url::to(['post/view', 'id' => 100, '#' => 'content']);
```

URL:

```
echo Url::to(['post/index'], true);
```

https URL:

```
echo Url::to(['post/index'], 'https');
```

`Url::to()` . . , admin post .

ID ():

```
echo Url::to(['index']); // -->> '/index.php?r=admin%2Fpost%2Findex'
```

():

```
echo Url::to(['post/index']); // -->> '/index.php?r=admin%2Fpost%2Findex'
```

():

```
echo Url::to(['/post/index']); // -->> '/index.php?r=post%2Findex'
```

URL:

```
echo Url::to();  
echo Url::to(['']);
```

GET URL [Url::current\(\)](#) .

```
$_GET = ['id' => 10, 'page' => 7], post/view.
```

URL :

```
echo Url::current(); // --> '/index.php?r=post%2Fview&id=10&page=7'
```

page URL :

```
echo Url::current(['page' => null]); // --> '/index.php?r=post%2Fview&id=10'
```

page URL :

```
echo Url::current(['page' => 12]); // --> '/index.php?r=post%2Fview&id=10&page=12'
```

URL : <https://riptutorial.com/ko/yii2/topic/5510/--url>

11:

Yii2 purpose . . .

Examples

.

- 1.
- 2.

: . Yii2 .

```
....
public function ValidateMyBusiness($attr, $params){
    // adding an error here means our validation is failed.
    if ($this->{$attr} > 1100) {
        $this->addError($attr, "Some error occurred");
    }
}
...
// calling above validator is simple as below:
public function rules(){
    return [
        ['money', 'validateMyBusiness', 'params' => ['targetAccount' => $this->account]];
    ]
}

# params array will be passed to our inline parameter as a second argument.
```

: <https://riptutorial.com/ko/yii2/topic/9187/--->

12:

Examples

yii2

```
use yii\web\Session;
```

```
$session = Yii::$app->session;  
$session->open(); // open a session  
$session->close(); // close a session
```

▪

```
$session = Yii::$app->session;  
  
$session->set('name', 'stack');  
OR  
$session['name'] = 'stack';  
OR  
$_SESSION['name'] = 'stack';
```

▪

```
$name = $session->get('name');  
OR  
$name = $session['name'];
```

```
$session->remove('name');  
OR  
unset($session['name']);  
OR  
unset($_SESSION['name']);  
  
$session->destroy(); // destroy all session
```

```
$session->removeAll();
```

```
$session->has('name')  
OR  
isset($session['name'])  
//both function return boolean value [true or false]
```

```
$session = Yii::$app->session;  
$session->setFlash('error', 'Error in login');
```



```
echo $session->getFlash('error');
```

```
$result = $session->hasFlash('error');
```

```
$session->removeFlash('error');
```

```
$session->removeAllFlashes();
```

```
\Yii::$app->session->set('name', 'stack');  
\Yii::$app->session->get('name');
```

```
\Yii::$app->getSession()->setFlash('flash_msg', 'Message');  
\Yii::$app->getSession()->getFlash('flash_msg');
```

.

```
$session = Yii::$app->session;  
$sess = $session['keys'];
```

.

```
$sess['first'] = 'abc';
```

.

```
$session['keys'] = $sess
```

URL

: () URL . , .

InvoiceController / actionUpdate :

```
Url::remember(Url::current(), 'returnInvoice');
```

ContactController / actionCreate :

```
if ($model->save()) {  
    $return = Url::previous('returnInvoice');  
    if ($return) {  
        return $this->redirect($return);  
    }  
    // ...  
}
```

URL .

InvoiceController / actionUpdate :

```
if ($model->save()) {  
    Url::remember(null, 'returnInvoice');  
    // ...  
}
```

- returnInvoice - .

: <https://riptutorial.com/ko/yii2/topic/3584/>

13:

Examples

Composer Extension

: . Composer .

- Github
- Open composer.json
- PSR-4 kmit/select2 .
- (: vendor/kmit/select2 .
- vendor/composer/autoload_psr4.php .

```
'kmit\\select2\\' => array($vendorDir . '/kmit/select2'),
```

- vendor/yiisoft/extensions.php .

```
'kmit/select2'(name of extension from composer.json file of extension) =>
array (
    'name' => 'kmit/select2',
    'version' => '1.0.0.0',
    'alias' => array (
        '@vendor/kmit/select2'(path of extension folder alias) => $vendorDir .
        '/kmit/select2' (path of extension folder),
    ),
),
```

Tutorial

: <https://riptutorial.com/ko/yii2/topic/2224/--->

14: API

Examples

API

countrylist

```
<?php

namespace app\models;

use Yii;

/**
 * This is the model class for table "countrylist".
 *
 * @property integer $id
 * @property string $iso
 * @property string $name
 * @property string $nicename
 * @property string $iso3
 * @property integer $numcode
 * @property integer $phonecode
 */
class Countrylist extends \yii\db\ActiveRecord
{
    /**
     * @inheritdoc
     */
    public static function tableName()
    {
        return 'countrylist';
    }

    /**
     * @inheritdoc
     */
    public function rules()
    {
        return [
            [['iso', 'name', 'nicename', 'phonecode'], 'required'],
            [['numcode', 'phonecode'], 'integer'],
            [['iso'], 'string', 'max' => 2],
            [['name', 'nicename'], 'string', 'max' => 80],
            [['iso3'], 'string', 'max' => 3]
        ];
    }

    /**
     * @inheritdoc
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
        ];
    }
}
```

```

        'iso' => 'Iso',
        'name' => 'Name',
        'nickname' => 'Nickname',
        'iso3' => 'Iso3',
        'numcode' => 'Numcode',
        'phonecode' => 'Phonecode',
    ];
}
}

```

restapi modelClass .

```

<?php
namespace app\controllers;
use yii\rest\ActiveController;
use Yii;
class CountrylistController extends ActiveController
{
    public $modelClass='app\models\Countrylist';
}
?>

```

restapi URL
URL .

```

'urlManager' => [
    'class' => 'yii\web\UrlManager',
    'enablePrettyUrl' => true,
    'showScriptName' => false,
    'rules' => [
        ['class'=>'yii\rest\UrlRule','controller'=>'countrylist']
    ],
],

```

API .

<http://localhost/countrylist> .

API Yii2

. ? . .

```

public function actions() {
    $actions = parent::actions();
    unset($actions['index']);
    return $actions;
}

public function actionIndex() {
    $activeData = new ActiveDataProvider([
        'query' => \common\models\Yourmodel::find(),
        'pagination' => false
    ]);
    return $activeData;
}

```

Content-Type

: () :

```
public function actionAsXML()
{
    $this->layout = false;
    Yii::$app->response->format = Response::FORMAT_XML;

    return ['aaa' => [1, 2, 3, 4]];
}
```

- FORMAT_HTML
- FORMAT_XML
- FORMAT_JSON
- FORMAT_JSONP
- FORMAT_RAW

text/plain **MIME** .

```
public function actionPlainText()
{
    $this->layout = false;
    Yii::$app->response->format = Response::FORMAT_RAW;
    Yii::$app->response->headers->add('Content-Type', 'text/plain');

    return $this->render('plain-text'); // outputs template as plain text
}
```

API : <https://riptutorial.com/ko/yii2/topic/6102/-api>

15:

- .
 - `public $depends = ['yii \ web \ YiiAsset', 'yii \ bootstrap \ BootstrapAsset', 'yii \ bootstrap \ BootstrapPluginAsset', 'cinghie \ fontawesome \ FontAwesomeAsset']];`

<https://github.com/yiisoft/yii2-app-advanced> .

cinghie adminLTE <https://github.com/cinghie/yii2-admin-lte> .

Examples

```
<?php
/* this example is based on the advanced template
 * This file is located in
 * backend/views/layouts/main.php
 */

use yii\helpers\Html;

// here the asset is registered
use cinghie\adminlte\AdminLTEAsset;
AdminLTEAsset::register($this);

?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
<html lang="<? = Yii::$app->language ?>">
<head>
    <meta charset="<? = Yii::$app->charset ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <? = Html::csrfMetaTags() ?>
    <title><? = Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>

<body class="hold-transition skin-blue sidebar-mini">

<?php $this->beginBody() ?>

<? = $content ?>

<?php $this->endBody() ?>

</body>
</html>
<?php $this->endPage() ?>
```

```

<?php

/**
 * This file is the Asset Bundle File located in
 * vendor/cinghie/yii2-admin-lte/AdminLTEAsset.php
 * @copyright Copyright &copy; Gogodigital Srls
 * @company Gogodigital Srls - Wide ICT Solutions
 * @website http://www.gogodigital.it
 * @github https://github.com/cinghie/yii2-admin-lte
 * @license GNU GENERAL PUBLIC LICENSE VERSION 3
 * @package yii2-AdminLTE
 * @version 1.3.10
 */

namespace cinghie\adminlte;

use yii\web\AssetBundle;

/**
 * Class yii2-AdminLTEAsset
 * @package cinghie\adminlte
 */
class AdminLTEAsset extends AssetBundle
{
    /**
     * @inherit
     */
    public $sourcePath = '@bower/';

    /**
     * @inherit
     */
    public $css = [
        'icons/css/ionsicons.css',
        'admin-lte/dist/css/AdminLTE.css',
        'admin-lte/dist/css/skins/_all-skins.css'
    ];

    /**
     * @inherit
     */
    public $js = [
        'admin-lte/dist/js/app.js'
    ];

    /**
     * @inherit
     */
    public $depends = [
        'yii\web\YiiAsset',
        'yii\bootstrap\BootstrapAsset',
        'yii\bootstrap\BootstrapPluginAsset',
        'cinghie\fontawesome\FontAwesomeAsset',
    ];
}

```

HTML


```
<!DOCTYPE html>
<html lang="en-EN">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="csrf-param" content="_csrf">
  <meta name="csrf-token"
content="M01tTVZLdlB1BQEqGyYcYHc5PwI1CRknfB1bBiAaPTNBfyk0Ehg8EQ==">
  <title>Profil</title>
  <link href="/assets/f3e48cde/css/bootstrap.css?v=1473788138" rel="stylesheet">
<link href="/assets/24e44190/css/font-awesome.css?v=1473866258" rel="stylesheet">
<link href="/assets/fa4335a5/ionicons/css/ionicons.css?v=1473866258" rel="stylesheet">
<link href="/assets/fa4335a5/admin-lte/dist/css/AdminLTE.css?v=1473866258" rel="stylesheet">
<link href="/assets/fa4335a5/admin-lte/dist/css/skins/_all-skins.css?v=1473866258"
rel="stylesheet"></head>
<body class="hold-transition skin-blue sidebar-mini">

....

</script><script src="/assets/69b4ffbe/jquery.js?v=1473788138"></script>
<script src="/assets/6aa8a809/yii.js?v=1473788138"></script>
<script src="/assets/f3e48cde/js/bootstrap.js?v=1473788138"></script>
<script src="/assets/fa4335a5/admin-lte/dist/js/app.js?v=1473866258"></script>

</body>
</html>
```

: <https://riptutorial.com/ko/yii2/topic/6900/>-

16:

HTTP

Examples

\yii\web\Cookie

```

$cookie = new Cookie([
    'name' => 'cookie_monster',
    'value' => 'Me want cookie!',
    'expire' => time() + 86400 * 365,
]);
\Yii::$app->getResponse()->getCookies()->add($cookie);

```

. PHP [setcookie](#) :

- name - .
- value -
- domain - .
- expire - **unix** .
- path - .
- secure - true, **HTTPS** .
- httpOnly - true **JavaScript** .

```

$value = \Yii::$app->getRequest()->getCookies()->getValue('my_cookie');

```

: (). JS / .

. example.com www.example.com . (: admin.example.com, profile.example.com) domain .

```

$cookie = new Cookie([
    'name' => 'cookie_monster',
    'value' => 'Me want cookie everywhere!',
    'expire' => time() + 86400 * 365,
    'domain' => '.example.com' // <<<=== HERE
]);
\Yii::$app->getResponse()->getCookies()->add($cookie);

```

example.com .

ID

autologin "remember me" . identityCookie .

identityCookie .

```

$config = [
    // ...
    'components' => [
        // ...
        'user' => [
            'class' => 'yii\web\User',
            'identityClass' => 'app\models\User',
            'enableAutoLogin' => true,
            'loginUrl' => '/user/login',
            'identityCookie' => [ // <---- here!
                'name' => '_identity',
                'httpOnly' => true,
                'domain' => '.example.com',
            ],
        ],
    ],
    'request' => [
        'cookieValidationKey' => 'your_validation_key'
    ],
    'session' => [
        'cookieParams' => [
            'domain' => '.example.com',
            'httpOnly' => true,
        ],
    ],
],
];

```

cookieValidationKey .

login logout session::cookieParams **samedomain** user::identityCookie . .

/admin **URL** .

```

$config = [
    // ...
    'components' => [
        // ...
        'session' => [
            'name' => 'admin_session',
            'cookieParams' => [
                'httpOnly' => true,
                'path' => '/admin',
            ],
        ],
    ],
],
];

```

: <https://riptutorial.com/ko/yii2/topic/2945/>

17:

Examples

:

```
composer global status
composer global require "codeception/codeception=~2.0.0" "codeception/specify="
"codeception/verify="
```

Faker :

```
cd /var/www/yii // Path to your application
composer require --dev yiisoft/yii2-faker:*
```

.

```
cd tests
codeception/bin/yii migrate
```

```
tests/codeception/config/config-local.php components['db'] .
```

```
/var/www/yii/vendor/bin .
```

```
.yml .
```

.. :

```
php -S localhost:8080
```

.

```
codecept run
```

:

- <http://www.yiiframework.com/doc-2.0/guide-test-environment-setup.html>
- <http://codeception.com/install>
- <https://github.com/yiisoft/yii2-app-basic/tree/master/tests>
- <https://github.com/yiisoft/yii2-app-advanced/tree/master/tests>

: Yii2 2.0.9 . 2.0.10 (Sam Dark) (). 2.0.10 2016 9 11 .

<https://github.com/yiisoft/yii2/milestones>

ActiveRecord

AR (PHPUnit).

```
$post = $this->getMockBuilder('\app\model\Post')
    ->setMethods(['save', 'attributes'])
    ->getMock();
$post->method('save')->willReturn(true);
$post->method('attributes')->willReturn([
    'id',
    'status',
    'title',
    'description',
    'text'
]);
```

catch ActiveRecord attributes () .

: <https://riptutorial.com/ko/yii2/topic/2226/>

18: Yii2 JQuery

Examples

max jquery

jquery . .

```
<?php
use yii\jui\DatePicker;
use yii\widgets\ActiveForm;
?>

<?php $form = ActiveForm::begin(['id' => 'profile-form']); ?>
.....
<?= $form->field($model, 'date_of_birth')->widget(DatePicker::classname(), ['dateFormat' =>
'php:M d, Y', 'options' => ['readonly' => true], 'clientOptions' => [ 'changeMonth' => true,
'changeYear' => true, 'yearRange' => '1980:'.date('Y'), 'maxDate' => '+0d']) ?>
.....
<?php ActiveForm::end(); ?>
```

jquery

/ .

```
<?php
use yii\jui\DatePicker;
use yii\widgets\ActiveForm;
?>

<?php $form = ActiveForm::begin(['id' => 'profile-form']); ?>
.....

<?php
$day = '+5d'; //if you want to display +5 days from current date means for future days.
#(or)
$day = '-5d'; //if you want to display -5 days from current date means older days.
?>
<?= $form->field($model, 'date_of_birth')->widget(DatePicker::classname(), ['dateFormat' =>
'php:M d, Y', 'options' => ['readonly' => true], 'clientOptions' => [ 'changeMonth' => true,
'changeYear' => true, 'yearRange' => '1980:'.date('Y'), 'maxDate' => $day]) ?>
.....
<?php ActiveForm::end(); ?>
```

jquery ()

from date date calendar calendar day date field . .

```
<?php $form = ActiveForm::begin(['id' => 'profile-form']); ?>
.....
<?= $form->field($model, 'from_date')->widget(DatePicker::classname(), ['dateFormat' => 'php:M
d, Y', 'options' => ['readonly' => true], 'clientOptions' => [ 'changeMonth' => true,
```

```
'changeYear' => true, 'yearRange' => '1980:'.date('Y'), 'onSelect' => new
yii\web\JsExpression('function(selected) { var dt = new Date(selected);
dt.setDate(dt.getDate() + 1); $("#filter-date-to").datepicker("option", "minDate", dt); }'))]]
?>

<?= $form->field($model, 'to_date')->widget(DatePicker::classname(), ['dateFormat' => 'php:MD, Y', 'options' => ['readonly' => true, 'id' => 'filter-date-to'], 'clientOptions' => [
'changeMonth' => true, 'changeYear' => true, 'yearRange' => '1980:'.date('Y')]]) ?>
.....
<?php ActiveForm::end(); ?>
```

Yii2 JQuery : <https://riptutorial.com/ko/yii2/topic/6366/--yii2-jquery->

19:

Examples

Yii UploadedFile [[yii \ web \ UploadedFile]] . [[yii \ widgets \ ActiveForm]] .

. . ,

```
namespace app\models;

use yii\base\Model;
use yii\web\UploadedFile;

class UploadForm extends Model
{
    /**
     * @var UploadedFile
     */
    public $imageFile;

    public function rules()
    {
        return [
            [['imageFile'], 'file', 'skipOnEmpty' => false, 'extensions' => 'png, jpg'],
        ];
    }

    public function upload()
    {
        if ($this->validate()) {
            $this->imageFile->saveAs('uploads/' . $this->imageFile->baseName . '.' . $this->imageFile->extension);
            return true;
        } else {
            return false;
        }
    }
}
```

imageFile . png jpg [[yii \ validators \ FileValidator]] file .upload() .

file , , MIME . .

: image .image [[yii \ validators \ ImageValidator]] .

.

```
<?php
use yii\widgets\ActiveForm;
?>

<?php $form = ActiveForm::begin(['options' => ['enctype' => 'multipart/form-data']]) ?>
```



```

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

<button>Submit</button>

<?php ActiveForm::end() ?>

```

Y enctype . fileInput() <input type="file"> .

: 2.0.8 [yii \ web \ widgets \ ActiveForm :: fileInput | fileInput] enctype .

```

namespace app\controllers;

use Yii;
use yii\web\Controller;
use app\models\UploadForm;
use yii\web\UploadedFile;

class SiteController extends Controller
{
    public function actionUpload()
    {
        $model = new UploadForm();

        if (Yii::$app->request->isPost) {
            $model->imageFile = UploadedFile::getInstance($model, 'imageFile');
            if ($model->upload()) {
                // file is uploaded successfully
                return;
            }
        }

        return $this->render('upload', ['model' => $model]);
    }
}

```

[[yii \ web \ UploadedFile :: getInstance ()]] UploadedFile .

file maxFiles . maxFiles 0 . max_file_uploads :20) PHP . upload() upload() .

```

namespace app\models;

use yii\base\Model;
use yii\web\UploadedFile;

class UploadForm extends Model
{
    /**
     * @var UploadedFile[]
     */
    public $imageFiles;

    public function rules()

```

```

    {
        return [
            [['imageFiles'], 'file', 'skipOnEmpty' => false, 'extensions' => 'png, jpg',
            'maxFiles' => 4],
        ];
    }

    public function upload()
    {
        if ($this->validate()) {
            foreach ($this->imageFiles as $file) {
                $file->saveAs('uploads/' . $file->baseName . '.' . $file->extension);
            }
            return true;
        } else {
            return false;
        }
    }
}

```

fileInput() multiple .

```

<?php
use yii\widgets\ActiveForm;
?>

<?php $form = ActiveForm::begin(['options' => ['enctype' => 'multipart/form-data']]) ?>

    <?= $form->field($model, 'imageFiles[]')->fileInput(['multiple' => true, 'accept' =>
    'image/*']) ?>

    <button>Submit</button>

<?php ActiveForm::end() ?>

```

, UploadedFile::getInstances() UploadedFile::getInstance() UploadedFile UploadForm::imageFiles
.

```

namespace app\controllers;

use Yii;
use yii\web\Controller;
use app\models\UploadForm;
use yii\web\UploadedFile;

class SiteController extends Controller
{
    public function actionUpload()
    {
        $model = new UploadForm();

        if (Yii::$app->request->isPost) {
            $model->imageFiles = UploadedFile::getInstances($model, 'imageFiles');
            if ($model->upload()) {
                // file is uploaded successfully
                return;
            }
        }
    }
}

```

```
        return $this->render('upload', ['model' => $model]);  
    }  
}
```

: [https://riptutorial.com/ko/yii2/topic/2221/-](https://riptutorial.com/ko/yii2/topic/2221/)

20:

Examples

Yii2

.

.

signup.php

(ID)

1. use yii\bootstrap\ActiveForm; use yii\bootstrap\ActiveForm; ()
2. use yii\widgets\ActiveForm; use yii\widgets\ActiveForm;
3. 'enableAjaxValidation' => true (ID).

```
<?php
use yii\bootstrap\ActiveForm;
use yii\widgets\ActiveForm;
?>

<?= $form->field($modelUser, 'email', ['enableAjaxValidation' => true])
    ->textInput(['class'=>'form-control', 'placeholder'=>'Email']); ?>
```

```
use yii\web\Response;use yii\widgets\ActiveForm;
```

```
<?php
use yii\web\Response;
use yii\widgets\ActiveForm;

.
.// Your code
.

public function actionSignup() {

    $modelUser = new User();

    //Add This For Ajax Email Exist Validation
    if(Yii::$app->request->isAjax && $modelUser->load(Yii::$app->request->post())){
        Yii::$app->response->format = Response::FORMAT_JSON;
        return ActiveForm::validate($modelUser);
    }
    else if ($model->load(Yii::$app->request->post())) {

    }

}

?>
```

```
[['email'],'unique','message'=>'Email already exist. Please try another one.'],
```

```
:
```

```
<?php
.....

    <?= $form->field($modelUser, 'email')->textInput(['class'=>'form-
control','placeholder'=>'Email']) ?>
.....
```

```
<?php
use yii\web\Response; // important lines
use yii\widgets\ActiveForm; // important lines

.
.// Your code
.

public function actionSignup()
{

    $modelUser = new User();

    //Add This For Ajax Validation
    if(Yii::$app->request->isAjax && $modelUser->load(Yii::$app->request->post())){
        Yii::$app->response->format = Response::FORMAT_JSON;
        return ActiveForm::validate($modelUser);
    }
    if ($modelUser->load(Yii::$app->request->post()) && $modelUser->save()) {
        return $this->redirect(['someplace nice']);
    }
    return $this->render('update', [
        'modelUser' => $modelUser,
    ]);
}
```

```
public function rules()
{
    return [
        .....
        [['email', 'unique', 'message'=>'Email already exist. Please try another one.'],
        .....
    ]
}
```

```
/
```

```
keyUp <div class="help-block"></div> textbox .
```

```
onKeyUp .
```

```
yii2/widgets/ActiveForm.php :
```

```

<?php

namespace yii\widgets;

use Yii;
use yii\base\InvalidCallException;
use yii\base\Widget;
use yii\base\Model;
use yii\helpers\ArrayHelper;
use yii\helpers\Url;
use yii\helpers\Html;
use yii\helpers\Json;

class ActiveForm extends Widget
{
    public $action = '';
    public $method = 'post';
    public $options = [];
    .
    .
    .
    public $validateOnSubmit = true;
    public $validateOnChange = true;
    public $validateOnBlur = true;
    public $validateOnType = false;

    .
    .
    .
}

```

```
$validateOnBlur true . .
```

```
<?php $form = ActiveForm::begin(['id' => 'register-form', 'validateOnBlur' => false]); ?>
```

```

class User extends \yii\db\ActiveRecord
{
    public static function tableName()
    {
        return 'user_master';
    }

    // define validation in rule() function
    public function rules()
    {
        return [
            [['email_id'], 'email'],
            [['first_name'], 'required', 'on'=>['create', 'update']], // create scenario
            [['email_id'], 'required', 'on'=> ['admin', 'create', 'update', 'forgotpassword']],
            [['mobile'], 'required', 'on'=>['admin', 'create', 'update']],
        ];
    }
}

```

```

public function actionCreate()
{
    $model = new User();
    $model->scenario="create"; // use create scenario, create scenario validaion applied in
    this model

}
public function actionUpdate()
{
    $model = new User();
    $model->scenario="update"; // use update scenario, update scenario validaion applied in
    this model
}

```

Yii2 2.0.4 EachValidator .

```

[
    // ... other rules
    ['userIDs', 'each', 'rule' => ['integer']],
]

```

['integer'] Yii2 . : ['integer', 'min' => 1337] . ID .

```

[
    ['myAttr', function($attribute, $params) {
        if (!is_array($this->$attribute)) {
            $this->addError($attribute, "$attribute isn't an array!");
        }
    }]
]

```

: <https://riptutorial.com/ko/yii2/topic/839/>

21:

AR , . () UPDATE .

AR, .

AR 100 OK. asArray () .

AR . AR . .

. .

Examples

```
Post::find()->all();  
// SELECT * FROM post
```

```
Post::findAll(condition);
```

ActiveRecord .

```
$model = User::find()  
    ->where(['id' => $id])  
    ->andWhere('status = :status', [':status' => $status])  
    ->all();
```

```
$model = User::find()  
    ->orderBy(['id'=>SORT_DESC])  
    ->all();
```

Or

```
$model = User::find()  
    ->orderBy(['id'=>SORT_ASC])  
    ->all();
```

Where

```
$postsGreaterThan = Post::find()->where(['>', 'created_at', '2016-01-25'])->all();  
// SELECT * FROM post WHERE created_at > '2016-01-25'
```

```
$postsLessThan = Post::find()->where(['<', 'created_at', '2016-01-25'])->all();  
// SELECT * FROM post WHERE created_at < '2016-01-25'
```

```
$postsNotEqual = Post::find()->where(['<>', 'created_at', '2016-01-25'])->all();  
// SELECT * FROM post WHERE created_at <> '2016-01-25'
```

```
$postsInArray = Post::find()->where(['id' => [1,2,3]])->all();
```



```
// SELECT * FROM post WHERE id IN (1,2,3)
```

```
$postsInBetween = Post::find()  
->where(['between', 'date', "2015-06-21", "2015-06-27" ])  
->all();
```

```
$postsWithNullTitle = Post::find()->where(['title' => null]);  
// SELECT * FROM post WHERE title IS NULL
```

```
$postsAND = Post::find()->where(['title' => null, 'body' => null]);  
// SELECT * FROM post WHERE title IS NULL AND body IS NULL
```

```
$postsAND = Post::find()->where(['OR', 'title IS NULL', 'body IS NULL']);  
// SELECT * FROM post WHERE title IS NULL OR body IS NULL
```

```
$postsNotEqual = Post::find()->where(['NOT', ['created_at'=>'2016-01-25']])->all();  
// SELECT * FROM post WHERE created_at IS NOT '2016-01-25'
```

```
$postsNestedWhere = Post::find()->andWhere([  
    'or',  
    ['title' => null],  
    ['body' => null]  
])->orWhere([  
    'and',  
    ['not', ['title' => null]],  
    ['body' => null]  
]);  
// SELECT * FROM post WHERE (title IS NULL OR body IS NULL) OR (title IS NOT NULL AND body IS  
NULL)
```

LIKE OPERATOR with filterWhereRecord

```
$title = 'test';  
$description = 'test';
```

i) andFilterWhere ()

```
$postLIKE = Post::find()->where(['user_id' => Yii::$app->user->getId()])->  
>andFilterWhere(['or', ['title' => $title, 'description' => $description]])->all();  
//SELECT * FROM post WHERE user_id = 2 AND (('title` LIKE '%test%') OR (`description` LIKE  
`%test%'))
```

ii) orFilterWhere ()

```
$postLIKE = Post::find()->where(['user_id' => Yii::$app->user->getId()])->orFilterWhere(['or',  
['title' => $title, 'description' => $description]])->all();  
//SELECT * FROM post WHERE user_id = 2 OR (('title` LIKE '%test%') OR (`description` LIKE  
`%test%'))
```

iii) filterWhere ()

```
$postLIKE = Post::find()->filterWhere(['AND', ['title' => $title, 'description' =>
$description]])->andWhere(['user_id' => Yii::$app->user->getId()])->all();
//SELECT * FROM post WHERE ((`title` LIKE '%test%') AND (`description` LIKE '%test%')) AND
user_id = 2
```

: filterWhere () filterWhere () all andwhere () orWhere () . filterWhere ()

ActiveRecord

```
<?php
namespace models;

use yii\db\ActiveRecord;
use yii\behaviors\TimestampBehavior;

class Post extends ActiveRecord
{
    public static function tableName()
    {
        return 'post';
    }

    public function rules() {
        return [
            [['created_at', 'updated_at'], 'safe'],
        ];
    }

    public function behaviors() {
        parent::behaviors();

        return [
            'timestamp' => [
                'class' => TimestampBehavior::className(),
                'attributes' => [
                    ActiveRecord::EVENT_BEFORE_INSERT => ['created_at', 'updated_at'],
                    ActiveRecord::EVENT_BEFORE_UPDATE => ['updated_at']
                ],
                'value' => date('Y-m-d H:i:s'),
            ]
        ];
    }
}
```

```
public function beforeSave($insert)
{
    if($this->isNewRecord){
        //When create
    }else{
        //When update
    }

    return parent::beforeSave($insert);
}
```

```

public function afterSave($insert, $changedAttributes )
{
    if($insert){
        //When create
    }else{
        //When update
    }
    return parent::afterSave($insert, $changedAttributes);
}

```

```
$customer = Customer::findOne(10);
```

```
$customer = Customer::find()->where(['id' => 10])->one();
```

```
$customer = Customer::find()->select('name,age')->where(['id' => 10])->one();
```

```
$customer = Customer::findOne(['age' => 30, 'status' => 1]);
```

```
$customer = Customer::find()->where(['age' => 30, 'status' => 1])->one();
```

id .

```
$model = User::findOne($id);
```

id .

```
$model = User::findOne($id)->name;
```

.

```
$model = User::find()->one(); // give first record
```

```
$model = User::find()->where(['id' => 2])->one(); // give single record based on id
```

.

```
$model = User::find()->select('name,email_id')->where(['id' => 1])->one();
```

```
$model = User::find()->select(['id','name','email_id'])->where(['id' => 1])->one();
```

```
$model = User::find()->select(['id','name','email_id'])->orderBy(['id' => SORT_DESC])->one();
```

OR

```
$model = User::find()->select(['id','name','email_id'])->orderBy(['id' => SORT_ASC])->one();
```

:

: 5 5 1 .

```
$subQuery = Post::find()->select(['customer_id'])->where(['status' => 2]); //fetch the
customers whos posts are inactive - subquery
$query = Customer::find()->where(['NOT IN', 'id', $subQuery])->all(); //Exclude the customers
whos posts are inactive by using subquery
```

: <https://riptutorial.com/ko/yii2/topic/1516/>-

S. No		Contributors
1	yii2	Bibek Lekhak , Community , Farcaller , jagsler , Mohan Rex , Muhammad Shahzad , Pasha Rumkin , Sam Dark , urmaul , Yasar Arafath , Yasin Patel , Yatin Mistry
2	Ajax	Anton Rybalko , Ilyas karim , meysam
3	Pjax	gmc , Manikandan S , Muaaz Rafi , Shaig Khaligli , yafater , Yasin Patel
4	Yii2 ActiveForm	Hina Vaja , Manikandan S , particleflux
5	Yii2 OAuth2 - : Facebook OAuth2	ThanhPV
6		mnoronha , Mohan Rex , Salem Ouerdani , Sam Dark , topher
7		Sam Dark , Yasin Patel
8		Ali MasudianPour , jlapoutre , Sam Dark
9		jagsler , Kiran Muralee
10	URL	IStranger , Ярослав Гойса
11		Ejaz Karim
12		Brett , Goke Obasa , jlapoutre , MikelG , Yasin Patel
13		Insane Skull , mnoronha , Sam Dark , vishuB
14	API	jagsler , jlapoutre , yafater , Yasin Patel
15		Thomas Rohde
16		IStranger , Sam Dark
17		Antonín Slejška , Bizley , Sam Dark , XzAeRo
18	Yii2 JQuery	Manikandan S
19		Sam Dark
20		jagsler , Mihai P. , Nana Partykar , Sam Dark , Yasin Patel
21		Insane Skull , Manikandan S , Michael St Clair , Mike Artemiev , particleflux , saada , Sam Dark , Yasar Arafath , Yasin Patel