

 無料電子ブック

学習

yii2

Free unaffiliated eBook created from
Stack Overflow contributors.

#yii2

.....	1
1: yii2	2
.....	2
.....	2
Examples.....	2
.....	2
Composer	2
Composer.....	3
Yii.....	3
.....	3
Yii2ubuntu.....	4
2: Ajax	7
Examples.....	7
Ajax.....	7
Ajax.....	8
3: Pjax	11
Examples.....	11
1.....	11
2.....	11
pjax.....	11
pjax.....	12
pjax.....	12
Pjax.....	12
4: Yii2 ActiveForm	14
Examples.....	14
Yii2.....	14
ActiveForm.....	15
5: Yii2 OAuth2 - Facebook OAuth2	17
Examples.....	17
Facebook.....	17
yii2-authclient.....	18

.....	19
Facebookredirect_url.....	20
onAuthSuccess.....	21
6:	22
.....	22
Examples.....	22
.....	22
Where.....	23
ActiveRecord.....	24
1.....	25
1.....	26
.....	27
7:	28
.....	28
Examples.....	28
.....	28
8:	29
.....	29
Examples.....	29
.....	29
.....	29
Cookie.....	29
.....	30
Cookie.....	31
9:	32
Examples.....	32
.....	32
.....	32
10:	34
Examples.....	34
yii2.....	34
.....	34

.....	34
.....	34
.....	34
.....	34
.....	35
.....	35
.....	35
.....	35
.....	36
URL.....	36
11:	38
Examples.....	38
Yii2.....	38
.....	39
orderBy.....	41
12:	43
Examples.....	43
.....	43
.....	43
.....	43
.....	44
.....	44
//.....	44
.....	45
.....	45
.....	45
.....	45
.....	45
.....	45
.....	45
.....	45
.....	46
.....	46
.....	46
13: Yii2 JQuery	47

Examples.....	47
maxjquery.....	47
jquery.....	47
jquery.....	47
14:	49
Examples.....	49
.....	49
ActiveRecord.....	50
15:	51
Examples.....	51
.....	51
.....	51
.....	51
.....	52
.....	52
.....	53
16: URL	55
.....	55
Examples.....	55
URL.....	55
17: API	57
Examples.....	57
API.....	57
APIYii2.....	58
Content-Type.....	59
18:	60
Examples.....	60
Composer.....	60
19:	61
Examples.....	61
Yii2.....	61
.....

/..... 63

..... 63

..... 64

20: **66**

..... 66

..... 66

Examples..... 66

..... 66

..... 67

HTML..... 68

21: **69**

Examples..... 69

..... 69

Web..... 69

Cookie..... 69

..... 70

..... **71**

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をいめる

YiiはなWebプログラミングフレームワークです。つまり、PHPをしてあらゆるのWebアプリケーションをするためにできます。コンポーネントベースのアーキテクチャとなキャッシュサポートにより、ポータル、フォーラム、コンテンツシステムCMS、プロジェクト、RESTful Webサービスなどのアプリケーションののにしています。

バージョン

バージョン	
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

インストールまたはセットアップ

Yii2は2つのでインストールできます。らです

1. Composerでのインストール
2. アーカイブファイルからのインストール

Composerでのインストール

Composerのインストール

まだComposerをインストールしていないは、getcomposer.orgの**にってください**。LinuxおよびMac OS Xでは、の**コマンド**をします。

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

Windowsの、に[composer-setup.exe](#)をダウンロードしてインストールしてください。github APIのアクセスをして、Github APIのレートを超えなければならないかもしれません。

Yiiのインストール

Composerがインストールされているは、Webアクセスなフォルダの**でのコマンド**をしてYiiをインストールできます。

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

の**コマンド**をしてYii2をテンプレートでインストールします。

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

なテンプレートでYii2をインストールするには

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

その、しいデータベースをし、それにしてcommon / config / main-local.phpのコンポーネント['db']をします。の**コマンド**をします。

```
php yii migrate
```

アーカイブファイルからのインストール

1. [Yii-download](#)からアーカイブファイルを**ダウンロード**する
2. ダウンロードしたファイルをWebアクセスなフォルダにします。
3. cookieValidationKeyのをしてconfig / web.phpファイルをします

なタイプのキーをできます

```
'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をubuntuにインストールする

まず、をインストールする必要があります。コンポーザーのインストール。

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

すぐディレクトリをする

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

のをする

```
composer
```

がインストールされました。

Yii2のインストールには2つのがあります。

1.アーカイブファイルからのインストール

のリンクからzipファイルをしてください。

/var/www/htmlディレクトリにします。

<https://github.com/yiisoft/yii2/releases/download/2.0.8/yii-advanced-app-2.0.8.tgz>

"advanced"フォルダのにします。でするか、のコマンドをしてください。

```
cd advanced
```

のコマンドをします。

```
php init
```

Composerをしたインストール

コンポーザでインストールするには、githubトークンが必要です。トークンについては、GitHubにサインアップするがあります。

サインアップの、トークンをすることができます

トークンをする

1. いずれかのページにあるプロフィールをクリックし、[]をクリックします。
2. ユーザーのサイドバーで、[アクセストークン]をクリックします。
3. [新しいトークンを]をクリックします。
4. トークンにわかりやすいをけます。
5. このトークンにするスコープをします。
6. [トークンの]をクリックします。
7. トークンをクリップボードにコピーします。セキュリティのから、このページからした、もトークンをもうることはできません。

リファレンス <https://help.github.com/articles/creating-an-access-token-for-command-line-use/>

トークンコピーをした

ディレクトリをする

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

のコマンドをする

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

```
composer config -g github-oauth.github.com f1eefb8f188c22dd6467f1883cb2615c194d1ce1
```

yii2をインストールする

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

"advanced"フォルダのにします。でするか、のコマンドをしてください。

```
cd advanced
```

のコマンドをします。

```
php init
```

しました

すぐすることができます。

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

そして

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

オンラインでyii2をいめるをむ <https://riptutorial.com/ja/yii2/topic/788/yii2をいめる>

2: Ajax リクエスト

Examples

Ajax フォームの

ファイルをする

```
<?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(); ?>
```

Javascript

```
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");
        });
    });
});
```

コントローラアクション

```

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
            ];
        }
    }
}

```

レンダリング **Ajax** ビュー

`Controller::renderAjax()` メソッドをして、**Ajax** リクエストにすることができます。このメソッドは、ビューにされている **JS / CSS** スクリプトとファイルをレンダリングにするをいて、`renderPartial` とています

ビューファイルにログインフォームがあるとします。

```

<?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>

```

コントローラーアクションで `renderPartial()` をうと

```

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

```

アクション

```

<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>

```

コントローラーアクションで `renderAjax()` をすると、のようになります。

```

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

```

アクション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>
```

のをビューからしたいデリゲートをぐため

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

アクション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>
```

オンラインでAjaxリクエストをむ <https://riptutorial.com/ja/yii2/topic/2944/ajaxリクエスト>

3: Pjax

Examples

ステップ1をする

ビュー \ 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(); ?>
```

ステップ2サーバーサイドコード

```
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のい

ビューのにこのをします。

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

ながなコンテンツのりにの2をします。

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

pjaxをリロードする

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

pjaxでタイムアウトをする

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

timeoutのをできます。これは、するミリデフォルトは1000です。サーバーのがこのタイムアウトよりきいは、フルページロードがトリガーされます。

デフォルトでは、pjaxはGETメソッドをしてフォームをします。ののように、フォームメソッドをPOSTにできます

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

Pjaxのない

Yii Framework 2.0には、ページみみをするJavaScriptライブラリPjaxがみまれています。これは、Ajaxによってされたページのをするだけでみます。これにより、ページにのくのアセットがあるになにつながります。いくつかのプロジェクトではこのをしており、したをしたいとえていました。

ページ1は、のしかまないなページです。ページ2には、ActiveFormとのウィジェットがまれています。インラインJavaScriptをするためには、ActiveForm JavaScriptリソースをロードするがありますが、Page 1にはこれらのアセットはまれていないため、アクティブフォームをしようとするとJavaScriptエラーがしました 'Uncaught TypeErrorundefined is a'。

すべてのページにロードされるアセットバンドルにActiveFormアセットをめ、エントリページでしいスクリプトをできるようにします。

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

のでは、ページ1にいくつかのウィジェットNavBarなどがまれています。ページ2には、じウィジェットと、さらにいくつかActiveFormなどがまれています。Pjaxをしてページをロードするとき、カスタムインラインJavaScriptがされていましたが、ActiveFormウィジェットによってされた

インラインスクリプトは、コードがしていないためしていませんでした。デバッグでは、ActiveForm initがされていましたが、'this'はActiveFormにしていないようです。これはにNavBar divにしています。IDをべると、ActiveFormはIDがw1であるとしていましたが、NavBarはそのページでにつかったウィジェットであるため、そのIDがりてられていました。

ウィジェットIDをするためにYiiにしないでください。わりに、ウィジェットをするときにIDをして、それらのIDのをしてください。

がされてからPjaxがに1,000ミリキャンセルされました。

Pjaxのタイムアウトをやしてください。デフォルトでは1にされています。これはサイトではされるはずですが、しかし、xdebugをしているは、ページのみみがこのをえています。

WebアプリケーションがPost-Redirect-GetPRGパターンをしています。Pjaxはリダイレクトだけでなくページをリロードします。

これはPjaxのされたです。リダイレクトはPjaxをしているときにそのをたさないため、がPjaxであるかどうかをし、そうであれば、リダイレクトするわりにコンテンツをレンダリングします。はのようになります。

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

PjaxとYiiのはですかもしあなたがかをしたか、たちよりもれたをっていれば、のコメントをください

オンラインでPjaxをむ <https://riptutorial.com/ja/yii2/topic/4554/pjax>

4: Yii2 ActiveForm

Examples

Yii2のフォームフィールド

しいのための**Yii2**のビューページのをす

これらはyii2 ActiveFormをしてフォームをするためにするがあるクラスです

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

のは、フォームのIDをすと、フォームにクラスをするをすをす、フォームのformタグをします。

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

ここで\$ modelこのにここにされているモデルオブジェクトを、するコントローラからしたフォームとバインドするデータベーステーブルフィールドをします。

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

'username'と 'password'は、がバインドされるテーブルフィールドのです。

ここでは、コードのにフォームのためのボタンをれ、ボタンテキストとなCSSクラスとして 'ログイン'をそれにします。

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

ここではコードのにフォームをクローズしています

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

パスワードフィールドの

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

TextFieldをする

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

しフォームフィールドを

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

ドリップダウンの

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

Idとをむドリップダウンリスト

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

FileUploaderをする

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

プレースホルダとカスタマイズされたラベルの

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

ActiveFormの

アクティブなでajaxとクライアントのを/にすることができます。

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

1. `enableClientValidation` は、デフォルトでActiveFormでになっています。フォームでクライアントのがないは、`false`としてりてることでにすることができます。
2. `enableAjaxValidation` は、ActiveFormではデフォルトでになっています。にするは、のようにActiveFormでするがあります。
3. `validationUrl` -あなたは、このフォームののコントローラのアクションですべてのコーディ

ングをしたいは、々がしてのActiveFormをすることができvalidationUrl。これをしなかつたは、フォームのアクションがされます。

の2つのはフォームにします。フォームののフィールドにしてのみajaxをチェックするは、そののフィールドにしてenableAjaxValidationをできます。フォームではなく、そのフィールドでのみします。

たとえば、フォームで、ユーザーをしたいは、フォームにしたユーザーのがすでにします。そのフィールドにしてこのenableAjaxValidationをできます。

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

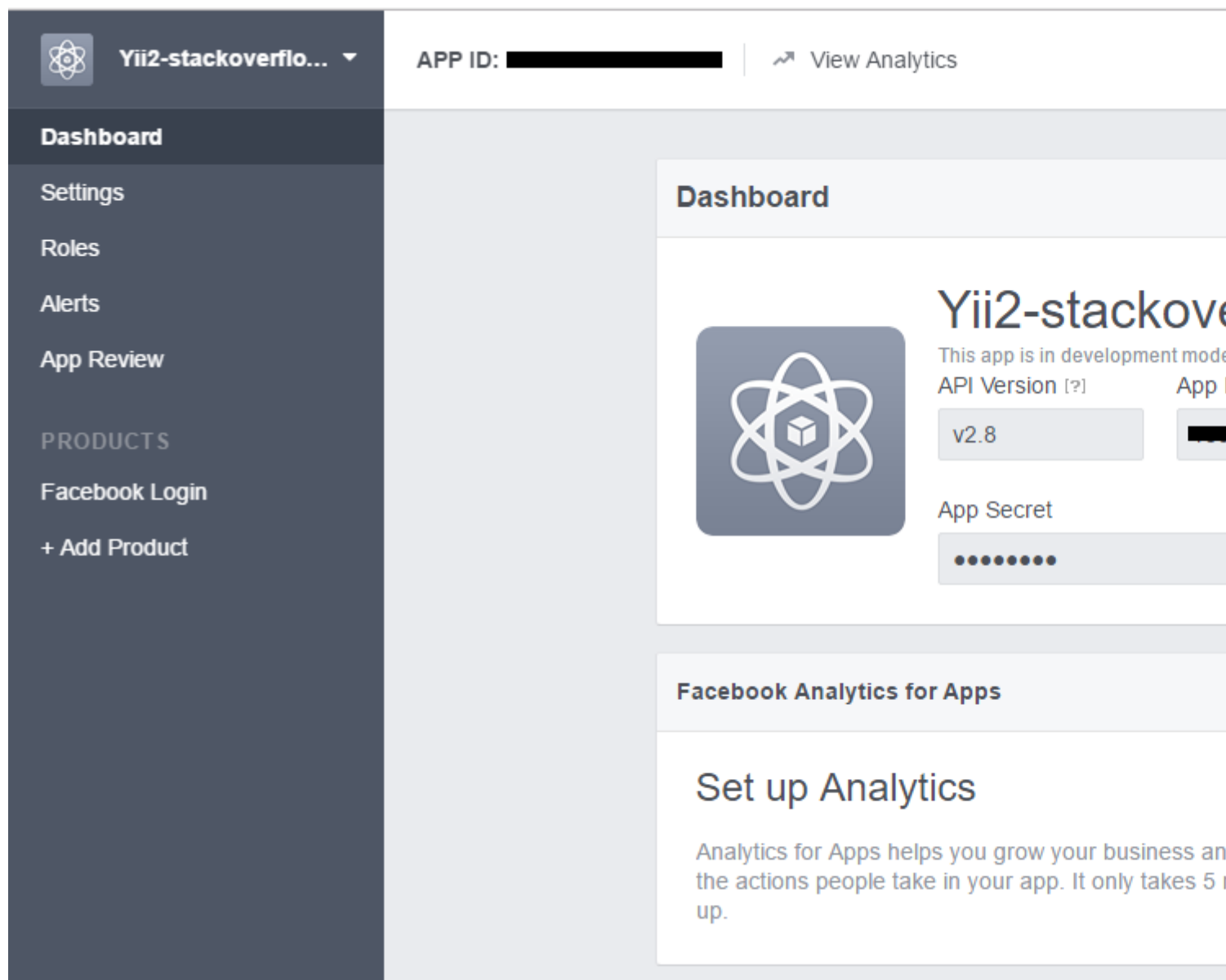
オンラインでYii2 ActiveFormをむ <https://riptutorial.com/ja/yii2/topic/6807/yii2-activeform>

5: Yii2 OAuth2 - の Facebook OAuth2

Examples

Facebookのにアプリをする

<https://developers.facebook.com/>にしてアプリをします。



[Add product を Add product クリックし、 Facebook Login をします

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 OAuth 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をインストールする

これをインストールするに、yii2-appをインストールする必要があります。ここでは、yii2-basicテンプレートをします。 [ここにインストールするためのガイド](#)。

る

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

またはする

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

あなたのcomposer.json requireセクションにcomposer.json。


```
        ],
    ];
}

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

私たちは `yii\authclient\AuthAction` を使って URL をし、Facebook のログインページにリダイレクトします。

`onAuthSuccess` は、ユーザーをし、アプリにログインするためにされました。

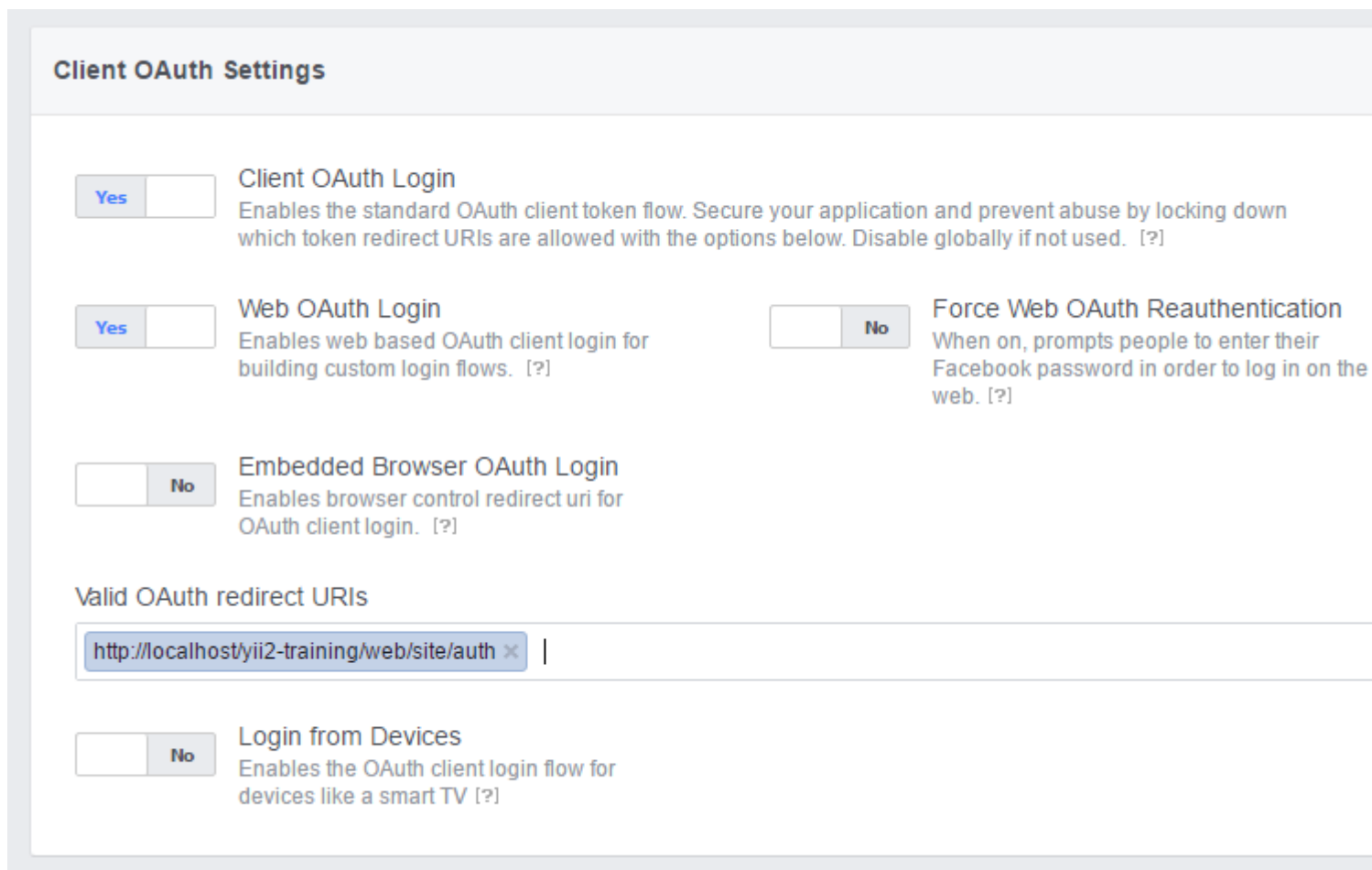
Facebook のアプリに `redirect_url` をする

`yii2-app` で `prettyUrl` をにすると、`redirect_uri` はのようになります

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

とかなりの URL をにする

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



Client OAuth Settings

Yes **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 **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の

```
/**
 * @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);
}
```

オンラインでYii2 OAuth2 - のFacebook OAuth2をむ <https://riptutorial.com/ja/yii2/topic/7428/yii2-oauth2-----のfacebook-oauth2>

6: アクティブレコード

ARは、1つのレコードを、またはするがあるにです。ダーティにされたもののみをするのサポートにより、データベースからのをにし、にのがじレコードをするのさまざまなのをらすUPDATEがられます。

アプリケーションになロジックがなく、エンティティをとしないは、ARが、にです。

ARは、1ページあたり100レコードのをもたらすなクエリにしてもOKです。これは、クエリビルダーやasArrayでされたをするのとじようにはしませんが、するほうがです。

ARはなクエリにはおめできません。これらは、データのまたはにするものであるため、されるものはARモデルにはあまりしません。これは、クエリビルダをするをおめします。

じことがインポートとエクスポートになります。のデータとによつてはなクエリがで、クエリビルダをするがです。

Examples

すべてのレコードを

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

または

キーまたはのによってアクティブなレコードモデルインスタンスをします。

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

ActiveRecordインスタンスのをします。

すべてをとに

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

orderByですべてを

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

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

Where

オペレーター

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

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

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

に

```
$postsInArray = Post::find()->where(['id' => [1,2,3]])->all();
// SELECT * FROM post WHERE id IN (1,2,3)
```

のに

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

ヌル

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

そして

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

または

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

NOT

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

ネストされた

```

$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 filterWhere アクティブなレコードメソッド

たとえば、フィルタでは、ログインしているユーザーがしたのタイトルやをしてをフィルタします。

```

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

```

iandFilterWhere

```

$postLIKE = Post::find()->where(['user_id' => Yii::$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%'))

```

iiorFilterWhere

```

$postLIKE = Post::find()->where(['user_id' => Yii::$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%'))

```

iiifilterWhere

```

$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をしているは、filterWhereのにall andwhereまたはorWhereをびすがあります。

イベントベースのフィールドをつActiveRecordクラスをする

```

<?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'),
            ],
        ];
    }
}

```

または、これを行うことができます

```

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);
}

```

レコードを1つつける

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

または

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

または

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

または

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

または

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

1つのクエリをする

idについてレコードをします。

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

idについてをします。

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

についてデータベースからのレコードをします。

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

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

についてデータベースからのレコードをします。

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

または

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

OrderBy

```
$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();
```


サブクエリをむアクティブなレコード

をできる。はのをできます。がをするとすぐに、はのレビューのにかれます。これで、サブクエリをしてすべてのアクティブなをつリストをするがあります。

おに5つのがある、5つののうちなくとも1つがのは、このをリストからするがあります。

```
$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
```

オンラインでアクティブレコードをむ <https://riptutorial.com/ja/yii2/topic/1516/アクティブレコード>

7: カスタム

き

Yii2には、プログラムのなををするにできるみみのバリデータがいくつかあります。しいビジネスロジックをするがあるは、のツールをするがあります。

Examples

の

めにバリデータのをしましょう

1. インラインバリデーター
2. スタンドアロンツール

インラインバリデータークラスにするバリデータのです。にはのメソッドとにしますが、Yii2によってされるなパラメータがあります。

```
....
public function ValidateMyBusiness($attr, $params){
    // adding an error here means our validation is failed.
    if ($this->{$attr} > 1100) {
        $this->addError($attr, "Some error ocured");
    }
}
...
// 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.
```

オンラインでカスタムをむ <https://riptutorial.com/ja/yii2/topic/9187/カスタム>

8: クッキー

CookieはHTTPリクエストのなので、コントローラでのをうことは、リクエストとレスポンスをにするがあります。

Examples

クッキーの

クッキーをするには、それをしてブラウザにするスケジュールをするには、しい\yii\web\Cookieクラスインスタンスをし、それをレスポンスクッキーコレクションにするがあります。

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

では、クッキークラスのコンストラクタにパラメータをしています。これらはにネイティブのPHP [setcookie](#)でされるものと同じです

- name - クッキーの。
- value - クッキーの。それがであることをしてください。ブラウザは、クッキーのバイナリデータにしていません。
- domain - あなたのためにクッキーをしているドメイン。
- expire - クッキーがにされるべきであることをすunixタイムスタンプ。
- path - クッキーがになるサーバーのパス
- secure - true、HTTPSがされているにのみCookieがされます。
- httpOnly - true、JavaScriptでCookieをできなくなります。

クッキーをむ

Cookieをみむには、のコードをします。

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

このコードでは、cookieコンポーネントをしてされたみりCookieをしていますデフォルトですべてのCookieにするため。したがって、JSコードをしてCookieを/すると、このメソッドをしてみることではできませんデフォルト。

サブドメインのCookie

セキュリティのから、デフォルトでは、Cookieはされたドメインと同じドメインでしかアクセスで

きません。たとえば、ドメイン `example.com` でCookieをした、ドメイン `www.example.com` Cookieをすることはできません。したがって、サブドメイン `admin.example.com`、`profile.example.com` などをするは、`domain`にするが`あり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);
```

これでcookieは `example.com` すべてのサブドメインからみることができ `example.com`。

クロスサブドメインとアイデンティティクッキー

ログインまたは「をえている」クッキーの、サブドメインクッキーのとじクオークがされます。しかし、は、ユーザーコンポーネントをし、`identityCookie`をのCookieにするがあります。

アプリケーションファイルをき、ユーザーコンポーネントに `identityCookie`パラメーターをします。

```
$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`はすべてのサブドメインでじにするがあります。

すべてのサブドメインの `login`と `logout` をにするために、`session::cookieParams` プロパティを `samedomain`を `user::identityCookie`としてするがあることにしてください。これは、のセクショ

ンでしくします。

セッションCookieパラメータ

1つのサブドメインからのサブドメインにアクセスしているにセッションをするがあるや、に/admin URLのにバックエンドアプリケーションをホストして々にセッションをしたいは、セッションCookieパラメータがです。

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

オンラインでクッキーをむ <https://riptutorial.com/ja/yii2/topic/2945/クッキー>

9: コンポーネント

Examples

アプリケーションコンポーネントの

コンポーネントの

- プロジェクトのルートフォルダに `components` というのフォルダをします。
- コンポーネントフォルダにコンポーネントをする `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` ファイルにコンポーネントをする

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

これでコンポーネントメソッドをできます

```
namespace app\controllers;

use Yii;

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

コンポーネントをしたドロップダウンリスト

MyComponent.phpでをする

```
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のコンポーネントをする

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

->あなたのでそれをう

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

オンラインでコンポーネントをむ <https://riptutorial.com/ja/yii2/topic/2217/コンポーネント>

10: セッション

Examples

yii2でのセッション

セッションクラスのインポート

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

セッションをする

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

セッションにをします。

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

セッションからをします。

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

セッションをする

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


すべてのセッションをする

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

セッションをチェックする

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

セッションフラッシュ

セッションフラッシュをする

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

セッションフラッシュをする

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

セッションフラッシュをする

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

セッションフラッシュを

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

すべてのセッションフラッシュをする

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

セッションをする

セッションのと

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

セッションフラッシュ

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

であるセッションのと

セッションをとします。

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

に、なをまたはします

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

にセッションにする

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

ですのためのURLをする

ユースケースなるするコントローラでしいレコードをしたにるには、のURLをえています。たとえば、のにするしいをします。

InvoiceController / actionUpdate

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

ContactController / actionCreate

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

したら、いしたURLをリセットすることができます

InvoiceController / actionUpdate

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

キーこのではreturnInvoiceはオプションです。

オンラインでセッションをむ <https://riptutorial.com/ja/yii2/topic/3584/セッション>

11: データベースの

Examples

Yii2 クエリービルダーをする

Yii2は、データベースからデータをするなをします。フィールド **emp_id**、**emp_name** および **emp_salary**をつなテーブルのをえてください。のをするために、クエリをします。

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

Yii2でのクエリをするには、たくさんのメソッドがあります。メソッドの1つは、**yii\db\Query** オブジェクトをすることです。

```
//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ループをして、**\$rows**のとのペアをループすることができます。

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

これはされます

Kiran、 25000

Midhun、 50000

Jishnu、 20000

Ajith、 25000

Akshay、 750000

よりくの

が25000にしいのをつけるがあるとします。

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

Yii2では、のクエリをするためのコード

```
$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();
```

が25000を超えるのをつけるがある、Yii2にコードをくことができます。

```
$rows=$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();
```

どこで

の**where**メソッドをしてのをできます。

```
// 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
```

のコードは、**kiran**と**salary 25000**というのをフェッチします。のがのをたす、call **one**はののみがフェッチされるようにします。すべてのをするには、**all**をするがあります。

allをすると、はにになります。が1つしかないでもこのには、としてのすべてのがまれています。レコードがしないはです。クエリ**one**がされないは、のをします。

sqlののコードをにします。

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

のクエリをYii2にきむのをにします。

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

ののセットは**and**をしてできます。でをするがあるにです。

のをするさらにのは、**where**メソッドのをすることです。のクエリは、のようにすることもできます。

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

ここでは、**'and'**をののとしてします。に、々はまた、「する」「きか」をする「または」「のの」、「きではない」、「のような」、「ないで」、「」、「いないの」「など」することができます、**'しません'**、**'>'**、**'<='**などがあります。

'in'と'like'の

が**20000**、**25000**、**50000**のをすがあるとします。のSQLでは、のようにクエリをします。

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

Yii2ではのようによくことができます。

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

じをするのはのとおりです。

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

が**20000,25000**、および**50000**でないすべてのをしたいは、に**'in'**を**'in'**のわりにできます。

にwhereので**'like'**をうをてみましょう。そのに「**gopal**」というをつすべてのをつけるがあるとしてます。はvenugopal、rajagopal、gopalakrishnanなどとすることができます。SQLクエリはのりです。

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

Yii2では、これを

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

そのに「**gopal**」と「**nair**」のをつすべてのをつけるがある。のようによくことができます

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

これは

emp_nameのように 'gopal'や 'nair'のようなからselect *

に、「**gopal**」と「**nair**」というがないすべてのをすには、「**きでない**」をできます。

orderByをすると、

orderByメソッドは、SQLクエリのORDER BYフラグメントをします。例えば、**emp_id**、**emp_first_name**、**emp_last_name**、**emp_salary**というフィールドをつテーブルをえてみましょう。ののでをべえるがあります。のようにsqlをします。

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

yii2では、のようクエリをできます

```
//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();
```

のをに、そしてをにべえるがあるは、のようプレーンSQLでくことができます。

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

のsqlは、のようyii2をしてビルドすることができます

```
//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();
```

のSQLをくときとじように、をしてORDER BYをすることもできます。たとえば、のようクエリをすることもできます。

```
//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をびして、をORDER BYフラグメントにすることができます。例えば

```
//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();
```

オンラインでデータベースのをむ <https://riptutorial.com/ja/yii2/topic/4167/データベースの>

12: データベースの

Examples

の

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

なnameには、しいについてにします。たとえば、でnewsというのしいテーブルをする、create_news_tableというをしてのコマンドをできます

```
yii migrate/create create_news_table
```

ファイルの

```
<?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()
    {
    }
    */
}
```

ドリップテーブル

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

```
}
```

すぐにテーブルフィールドをする

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

する

```
/**
 * 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');
    }
}
```

テーブルの

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

ドロップの

```
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'));
}
```

を

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

の

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

トランザクションマイグレーション

```
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');
}
```

トランザクションのをするさらには、コードを `safeUp()` および `safeDown()` メソッドに `safeDown()` ことです。これら2つのメソッドは、トランザクションでにまれているで `up()` および `down()` となります。として、これらのメソッドでのがした、すべてののはにロールバックされます。

のデータベースの

デフォルトでは、はdbアプリケーションコンポーネントによってされたじデータベースにされます。それらをのデータベースにしたいは、にすようなdbコマンドラインオプションをすることができます

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

のやりし

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

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

の

```
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
```

の

```
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
```

の

```
yii migrate
```

このコマンドはこれまでにされていないすべてのをします。これらのをすることをした、しいクラスごとにタイムスタンプのにupまたはsafeUpメソッドがされます。いずれかのがした、コマンドはりのをせずにします。

```
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
```

オンラインでデータベースのをむ <https://riptutorial.com/ja/yii2/topic/1929/データベースの>

13: テキストフィールドのYii2 JQueryカレンダー

Examples

のとしてmaxをつテキストフィールドのjqueryカレンダーをする

カレンダーでのとしてのをできるエンドユーザーのjqueryカレンダーをする。このシナリオでは、のコードがです。

```
<?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(); ?>
```

ののテキストフィールドにjqueryカレンダーをする

ののをし、ののをにするのあるフォームでは、このシナリオがちます。

```
<?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(); ?>
```

からまでのjqueryカレンダーを

とのカレンダーをしたいは、カレンダーのにもにフィールドのよりもきくなります。

```
<?php $form = ActiveForm::begin(['id' => 'profile-form']); ?>
.....
<?=$form->field($model, 'from_date')->widget(DatePicker::classname(), ['dateFormat' => 'php:Md, 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(); ?>
```

オンラインでテキストフィールドのYii2 JQueryカレンダーをむ

<https://riptutorial.com/ja/yii2/topic/6366/テキストフィールドのyii2-jqueryカレンダー>

14: テスト

Examples

テストの

コードのインストール

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

Fakerをインストールする

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

テストにのみされるデータベースをします。のデータベースをしたり、をすることができます。

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

tests/codeception/config/config-local.phpのcomponents['db']をして
tests/codeception/config/config-local.php。

/var/www/yii/vendor/binディレクトリをパスにします。

すべてのファイルと.ymlファイルをしてください。

ウェブサーバーをします。

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

テストをします。

```
codecept run
```

しくは

- <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>

これらは、Yii2バージョン2.0.9でです。バージョン2.0.10では、Sam Darkのテストのされたそ

してをすることがあります。バージョン2.0.10は、2016911にリリースされるはずですが。 <https://github.com/yiisoft/yii2/milestones>

ActiveRecordをモックする

データベースにしようとしなないARをモックしたいは、のようになしますPHPUnitをしている

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

このキャッチは、デフォルトでActiveRecordがしようとしているデータベーススキーマからリストをしているので、attributesメソッドをオーバーライドするがあることです。

オンラインでテストをむ <https://riptutorial.com/ja/yii2/topic/2226/テスト>

15: ファイルのアップロード

Examples

どうやってするの

ファイルのアップロード

Yiiでファイルをアップロードするには、アップロードされたファイルを`UploadedFile`オブジェクトとしてカプセルする`[[yii\web\UploadedFile]]`のけをりてうのがです。 `[[yii\widgets\ActiveForm]]`とモデルをみわせることで、なファイルアップロードメカニズムをにできます。

モデルの

プレーンテキストでするとに、1つのファイルをアップロードするには、モデルクラスをし、モデルのをしてアップロードされたファイルインスタンスをします。ファイルのアップロードをするためのルールもするがあります。えは、

```
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;
        }
    }
}
```

のコードでは、`imageFile` はアップロードされたファイルインスタンスをするためにされます。これは[[yii\validators\FileValidator]]をして、`png` または `jpg` ファイルがアップロードされていることをする `file` にけられてい `file`。 `upload()` メソッドはをし、アップロードされたファイルをサーバーにします。

`file`バリデーターをすると、ファイルの、サイズ、MIMEタイプなどをできます。については、コアバリデーターのセクションをしてください。

ヒントイメージをアップロードするは、わりに `image` バリデータをするをしてくださ。 `image` バリデーターは、[[yii\validators\ImageValidator]]をしてされ、がなイメージをけたかどうかをします。なイメージは、イメージをしてまたはできます。

レンダリングファイルの

に、ビューにファイルをします。

```
<?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() ?>
```

ファイルをにアップロードできるように、フォームに `enctype` オプションをするをえておくことがです。 `fileInput()` びしは、ユーザーがアップロードするファイルをできるようにする `<input type="file">` タグをレンダリングします。

ヒントバージョン2.0.8、[[yii\web\widgets\ActiveField::fileInput|fileInput]]は、ファイルフィールドがされるときににフォームに `enctype` オプションをします。

コントローラのアクションで、ファイルアップロードをするためのモデルとビューをびつけるコードをいてください

```
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]);
    }
}

```

このコードでは、フォームがされると、[[yii\web\UploadedFile::getInstance]]メソッドが呼びされ、アップロードされたファイルをUploadedFileインスタンスとしてします。に、モデルのをして、アップロードされたファイルがであることをし、ファイルをサーバーにします。

のファイルのアップロード

また、のファイルのをにアップロードすることもできます。のでしたコードをいくつかします。

まず、アップロードするファイルのをするために、fileにmaxFilesオプションをして、モデルクラスをするがあります。maxFilesを0すると、にアップロードできるファイルにがないことをします。にアップロードできるファイルのは、PHPディレクティブのmax_file_uploads デフォルトは20 upload()されていますmax_file_uploads upload()メソッドは、アップロードされたファイルを1つずつするようにするがあります。

```

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;
        }
    }
}

```

```
    }  
  }  
}
```

ビューファイルでは、`fileInput()` に `multiple` オプションをして、ファイルアップロードフィールドでのファイルができるようにするがあります。

```
<?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() ?>
```

そしてにコントローラのアクションでは、あなたが必ずあります `UploadedFile::getInstances()` のわり `UploadedFile::getInstance()` のりてるため `UploadedFile` にインスタンス `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]);  
    }  
}
```

オンラインでファイルのアップロードをむ <https://riptutorial.com/ja/yii2/topic/2221/ファイルのアップロード>

16: ルーティングとURL

すべてのURLは、ヘルパー`yii\helpers\Url`でするがあります。urlManagerでurlルールをすることをした、くのちます。

Examples

URLの

ヘルパー`yii\helpers\Url`は、URLをするメソッドのセットをします。このヘルパーは、ビュー/コントローラのコードでされることがあります。

ルートへのURL

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

パラメータきのルートへのURL

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

アンカーURL

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

URL

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

httpsスキームをしたURL

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

`Url::to()`メソッドにされるルートはコンテキストです。のモジュールとコントローラをすることがあります。たとえば、のモジュールが`admin`で、のコントローラが`post`ます。

アクションIDのみのルートスラッシュはまれていません

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

ルートにスラッシュはありません

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

ルートスラッシュでまる

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

されているURL

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

のルートと**GET**パラメータについてURLをするには、 `Url::current`をします。

`$_GET = ['id' => 10, 'page' => 7]`とすると、のは `post/view`です。

のURL

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

`page`パラメータなしのURL

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

された `page`パラメータをつのURL

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

オンラインでルーティングとURLをむ <https://riptutorial.com/ja/yii2/topic/5510/ルーティングとurl>

17: らかなAPI

Examples

りのAPIからめる

たちはをむテーブルをとっていますので、countrylistモデルとばれるモデルをします

```
<?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',
    ];
}
}

```

はrestapiのコントローラをし、モデルクラスのmodelClassをします。

```

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

```

restapiをうためにはかなりのURLがです
 たちはこのルールをpretty urlにします

```

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

```

その、たちはアクセスして、りのAPIをテストすることができます

<http://localhost/countrylist>はにのリストをえます。

りのAPIのデフォルトアクションをオーバーライドするYii2

として、デフォルトのインデックスアクションでページネーションをにし、すべてのをindexでしたいとします。どうやってそれができるそれはです。コントローラのインデックスアクションをのようにオーバーライドするがあります。

```

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;  
}
```

のアクションのContent-Typeをきする

ユースケースそのままのテキストのコンテンツをすアクションは1つだけです

```
public function actionAsXML()  
{  
    $this->layout = false;  
    Yii::$app->response->format = Response::FORMAT_XML;  
  
    return ['aaa' => [1, 2, 3, 4]];  
}
```

みのはのとおりです。

- FORMAT_HTML
- FORMAT_XML
- FORMAT_JSON
- FORMAT_JSONP
- FORMAT_RAW

text/plain MIMEタイプはそのままでもそのままですが、わりにこれをしてください

```
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  
}
```

オンラインでらかなAPIをむ <https://riptutorial.com/ja/yii2/topic/6102/らかなapi>

18: でエクステンションをインストールする

Examples

Composerなしでエクステンションをインストールする

Composerをすることをくおめします。のは、にComposerがあなたのためにうことで
す。

- Githubからなバージョンのアーカイブファイルをダウンロードする
- Open composer.json
- PSR-4 ロードセクションをつけて、例えば kmit/select2
- vendor/kmit/select2 ような vendor/kmit/select2 フォルダのするフォルダにファイルをします
。
- vendor/composer/autoload_psr4.php にのコードをしてください

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

- vendor/yiisoft/extensions.php にのコードをしてください

```
'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),  
    ),  
) ,
```

ビデオTutorial

オンラインででエクステンションをインストールするをむ <https://riptutorial.com/ja/yii2/topic/2224/>
でエクステンションをインストールする

19:

Examples

Yii2のデータベースからのをする

のがテキストボックスにされている、エラーメッセージがされないというはほとんどありません。

したがって、として、ユーザーにのメールをさせることはできません。

signup.php

あなたがのメールIDなしでしいユーザーにサインアップしたいページ

1. use yii\bootstrap\ActiveForm; し use yii\bootstrap\ActiveForm; **する**
2. use yii\widgets\ActiveForm; **をしてください** use yii\widgets\ActiveForm;
3. 'enableAjaxValidation' => true **をし** 'enableAjaxValidation' => true **このフィールドには、の**
メールIDをするユーザーをします。

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

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

コントローラ

これらのをに use yii\web\Response; use yii\widgets\ActiveForm;

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

.
.
.
.
.

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())) {
```

```
}  
  
}  
?>
```

モデル

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

データベースからのものの

のがされているにされないエラーメッセージにするがあります。たとえば、はこのメールでユーザーサインアップをしていません。

ビュー

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

コントローラ

```
<?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,  
    ]);  
}
```

モデル

```
public function rules()
```

```

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

```

フォーカス/キーアップでエラーメッセージをにする

デフォルトでは、*keyUp*の<div class="help-block"></div> textbox、またはがたされていないは *submit* ボタンをしたにエラーメッセージがされます。

によっては、にのみメッセージがながあります。つまり、 *onKeyUp* イベントでははわれません。

`yii2/widgets/ActiveForm.php` ファイルをチェックしましょう

```

<?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` はデフォルトで `true` されています。フレームワークファイルをすることはいいことです。そのため、フォームをするには、フレームワークをきするがあります。

```

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

```

のシナリオ

シナリオをすると、なるでをできます

モデルクラスでシナリオをする

```
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']],
        ];
    }
}
```

コントローラでシナリオをする

```
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
}
```

の

Yii2バージョン2.0.4、のをするためにされるEachValidatorがあります。

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

['integer']は、Yii2がするのすべてのバリデータオブジェクトであり、バリデータのをできます。 ['integer', 'min' => 1337]。ユーザーIDにがまれていない、ルールのはします。

のをわずににがまれているかどうかだけをしたいは、のバリデータをすることができます。

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

オンラインでもむ <https://riptutorial.com/ja/yii2/topic/839/>

20:

- はこののののでロードされます
 - `public $ depends = ['yii \ web \ YiiAsset', 'yii \ bootstrap \ BootstrapAsset', 'yii \ bootstrap \ BootstrapPluginAsset', 'cinghie \ fontawesome \ FontAwesomeAsset'];`

これは、なテンプレート <https://github.com/yiisoft/yii2-app-advanced> についています

こののcinghieアセットは、adminLTE <https://github.com/cinghie/yii2-admin-lte> のアセットパッケージです

Examples

これはレイアウトファイルのです

```
<?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() ?>
```


これはファイルです

```
<?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 = [
        'ionicons/css/ionicons.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',
    ];

}
```

されたをロードしてされた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>
```

オンラインでをむ <https://riptutorial.com/ja/yii2/topic/6900/>

21: なプロジェクトテンプレート

Examples

ホスティングでの

なプロジェクトテンプレートをホスティングにデプロイするのは、ホスティングWebサーバーがサポートしていない2つのWebルートがあるため、なプロジェクトテンプレートをなものよりもややしくしています。たちは、フロントエンドのURLがされるので、ディレクトリをするがあります `//site.localHTTP` およびバックエンドURLがなる `//site.local/admin` `まずhttp`。

エントリスクリプトをのWebルートにする

まず、Webrootディレクトリがです。しいディレクトリをし、ホスティングのWebルート `www` や `public_html` などとするをけます。に、 `www` がしたWebルートディレクトリをホスティングしているのをします。

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

wwwはフロントエンドディレクトリになりますので、 **frontend / web**のをそのディレクトリにしてください。 バックエンド/**Web**のを **www / admin**にします。いずれのもの、 **index.php**と**index-test.php**のパスをするがあります。

セッションとCookieをする

もともとバックエンドとフロントエンドはなるドメインですることをしています。すべてをじドメインにすると、フロントエンドとバックエンドはじCookieをし、をります。これをするには、のようにバックエンドアプリケーションの **config backend / config / main.php**をします。

```
'components' => [
  'request' => [
    'csrfParam' => '_csrf-backend',
    'csrfCookie' => [
      'httpOnly' => true,
      'path' => '/admin',
    ],
  ],
],
'user' => [
  'identityClass' => 'common\models\User',
```


クレジット

S. No		Contributors
1	yii2をいめる	Bibek Lekhak , Community , Farcaller , jagsler , Mohan Rex , Muhammad Shahzad , Pasha Rumkin , Sam Dark , urmaul , Yasar Arafath , Yasin Patel , Yatin Mistry
2	Ajaxリクエスト	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 - の Facebook OAuth2	ThanhPV
6	アクティブレコード	Insane Skull , Manikandan S , Michael St Clair , Mike Artemiev , particleflux , saada , Sam Dark , Yasar Arafath , Yasin Patel
7	カスタム	Ejaz Karim
8	クッキー	IStranger , Sam Dark
9	コンポーネント	Sam Dark , Yasin Patel
10	セッション	Brett , Goke Obasa , jlapoutre , MikelG , Yasin Patel
11	データベースの	jagsler , Kiran Muralee
12	データベースの	Ali MasudianPour , jlapoutre , Sam Dark
13	テキストフィールドのYii2 JQueryカレンダー	Manikandan S
14	テスト	Antonín Slejška , Bizley , Sam Dark , XzAeRo
15	ファイルのアップロード	Sam Dark
16	ルーティングとURL	IStranger , Ярослав Гойса
17	らかなAPI	jagsler , jlapoutre , yafater , Yasin Patel

18	でエクステンション をインストールする	Insane Skull , mnoronha , Sam Dark , vishuB
19		jagsler , Mihai P. , Nana Partykar , Sam Dark , Yasin Patel
20		Thomas Rohde
21	なプロジェクトテン プレート	mnoronha , Mohan Rex , Salem Ouerdani , Sam Dark , topher