AJAX-dependent drop-down list

Often, you’ll need a form with two dropdowns, and one dropdown’s values will be dependent on the value  
of the other dropdown. Using Yii’s built-in AJAX functionality, you can create such a dropdown.

Getting ready

1. Create a new application by using composer, as described in the official guide at  
   [http://www.yiiframework.com/doc-2.0/guide-start-insta]lation.html](http://www.yiiframework.com/doc-2.0/guide-start-installation.html)**.**
2. Create an @app/model/Product.php model as follows:

<?php

namespace app\models;  
use yii\db\ActiveRecord;  
class Product extends ActiveRecord  
{

public function rules()

{

return [

['title ', 'string'],

[['title', 'category\_id', 'sub\_category\_id'], 'required'],  
['category\_id', 'exist', 'targetAttribute' => 'id', 'targetClass' =>

'app\models\Category ' ],

['sub\_category\_id', 'exist', 'targetAttribute' => 'id', 'targetClass'

=> 'app\models\Category'],

];

}

public function attributeLabels()

{

return [

'category\_id' => 'Category',

'sub\_category\_id' => 'Sub category',

]; }

}

1. Create an @app/models/Category.php model as follows:

<?php

namespace app\models;  
use yii\db\ActiveRecord;  
class Category extends ActiveRecord  
{

public function rules()

{

return [

['title ', 'string'],

];

}

/\*\*

\* @return array  
\*/

public static function getSubCategories($categoryId)

{

$subCategories = [];  
if ($categoryId) {

$subCategories = self::find()

->where(['category\_id' => $categoryId])

->asArray()

->all();

}

return $subCategories;

}

}

1. Create a create\_category\_and\_product\_tables migration with the following command:  
   ./yii migrate/create create\_category\_and\_product\_tables
2. Update the just-created migration’s methods and list of imported classes as follows:

<?php

use yii\db\Schema;  
use yii\db\Migration;

class m150813\_005030\_create\_categories extends Migration  
{

public function up()

{

$tableOptions = null;

$this->createTable('{{%product}}', [

'id' => Schema::TYPE\_PK,

'category\_id' => Schema::TYPE\_INTEGER . ' NOT NULL',

'sub\_category\_id' => Schema::TYPE\_INTEGER . ' NOT NULL',

'title' => Schema::TYPE\_STRING . ' NOT NULL',

], $tableOptions);

$this->createTable('{{%category}} ', [

'id' => Schema::TYPE\_PK,

'category\_id' => Schema::TYPE\_INTEGER,

'title' => Schema::TYPE\_STRING . ' NOT NULL',

], $tableOptions);

$this->addForeignKey('fk\_product\_category\_id', '{{%product}}',

'category\_id', '{{%category}}', 'id');

$this->addForeignKey('fk\_product\_sub\_category\_id', '{{%product}}',

'category\_id', '{{%category}}', 'id');

$this->batchInsert('{{%category}}', ['id', 'title'], [

[1, 'TV, Audio/Video'],

[2, ' Photo'],

[3, 'Video']

]);

$this->batchInsert('{{%category}}', ['category\_id', 'title'], [

[1, 'TV'],

[1, 'Acoustic System'],

[2, 'Cameras'],

[2, 'Flashes and Lenses '],

[3, 'Video Cams'],

[3, 'Action Cams'],

[3, 'Accessories']

]);

}

public function down()

{

$this->dropTable('{{%product}}');

$this->dropTable('{{%category}}');

}

}

How to do it...

1. Create a controller file, @app/controllers/DropdownController . php, as follows:

<?php

namespace app\controllers;  
use app\models\Product;  
use app\models\Category;

use app\models\SubCategory;  
use Yii;

use yii\helpers\ArrayHelper;  
use yii\helpers\Json;  
use yii\web\Controller;  
use yii\web\HttpException;

class DropdownController extends Controller  
{

public function actionGetSubCategories($id)

{

if (!Yii::$app->request->isAjax) {

throw new HttpException(400, 'Only ajax request is allowed.');

}

return Json: :encode(Category::getSubCategories($id));

}

public function actionIndex()

{

$model = new Product();

if ($model->load(Yii::$app->request->post()) && $model->validate()) {  
Yii::$app->session->setFlash('success',

'Model was successfully saved'

);

}

return $this->render('index', [

'model' => $model,

]);

}

}

2. Create a view file, @app/views/dropdown/index.php, as follows:

<?php

use yii\bootstrap\ActiveForm;  
use yii\helpers\Html;  
use yii\helpers\Url;  
use app\models\Category;  
use yii\helpers\ArrayHelper;  
use yii\web\View;

$url = Url::toRoute(['dropdown/get-sub-categories']);

$this->registerJs("

(function() {

var select = $('#product-sub\_category\_id');  
var buildOptions = function(options) {  
if (typeof options === 'object') {

select. children('option' ).remove();

$('<option />')

.appendTo(select)

.html('Select a sub category')

$.each(options, function(index, option) {

$('<option />', {value:option.id})

.appendTo(select)

.html(option.title);

});

}

};

var categoryOnChange = function(category\_id){

$.ajax({

dataType : ' j son',

url: '" . $url . "&id=' + category\_id ,  
success: buildOptions

});

};

window.buildOptions = buildOptions;  
window.categoryOnChange = categoryOnChange;

})();

", View::POS\_READY);

?>

<h1>Product</h1>

<?php if (Yii::$app->session->hasFlash('success')): ?>

<div class="alert alert-success"><?= Yii::$app->session->getFlash('success'); ?  
></div>

<?php endif; ?>

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

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

<?= $form->field($model, 'category\_id')->dropDownList(ArrayHelper::map(

Category::find()->where('category\_id IS NULL')->asArray()->all(),'id',  
'title'), [

'prompt' => 'Select a category',

'onChange' => 'categoryOnChange($(this).val());',

]) ?>

<?= $form->field($model, 'sub\_category\_id')->dropDownList(

ArrayHelper::map(Category::getSubCategories($model->sub\_category\_id), 'id'

, 'title'), [

'prompt' => 'Select a sub category',

]) ?>

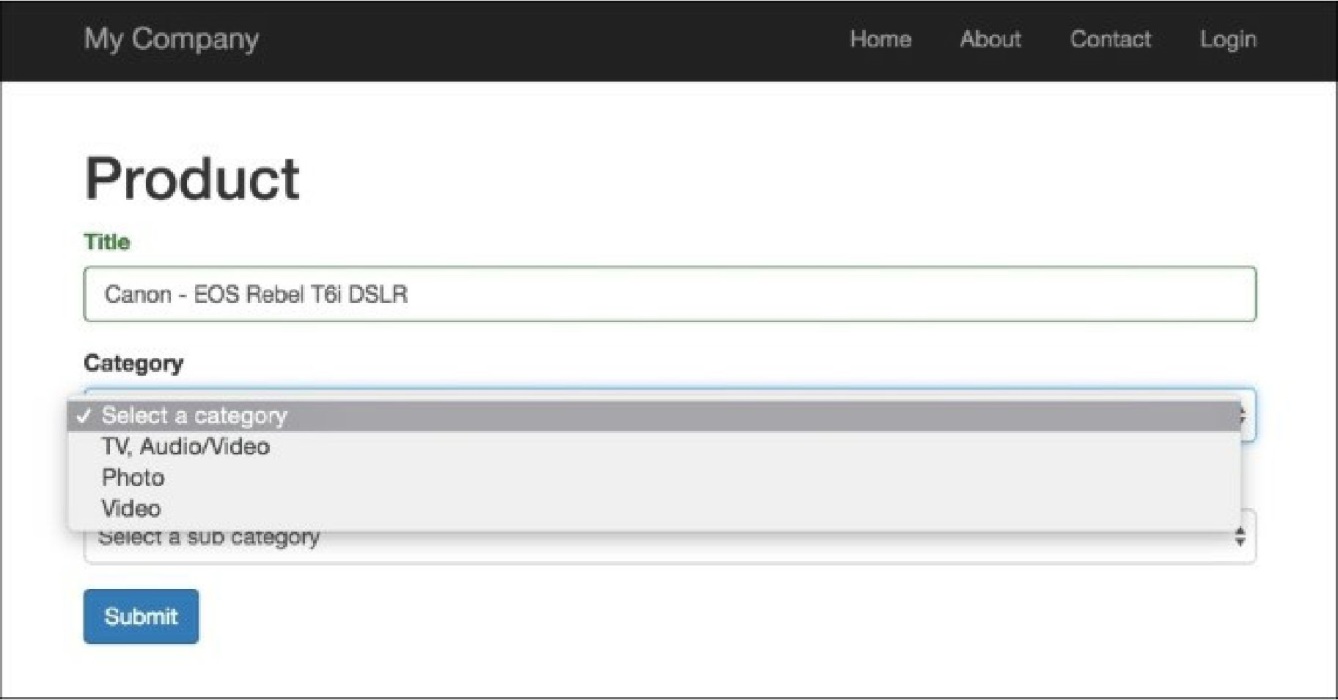
<div class="form-group">

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

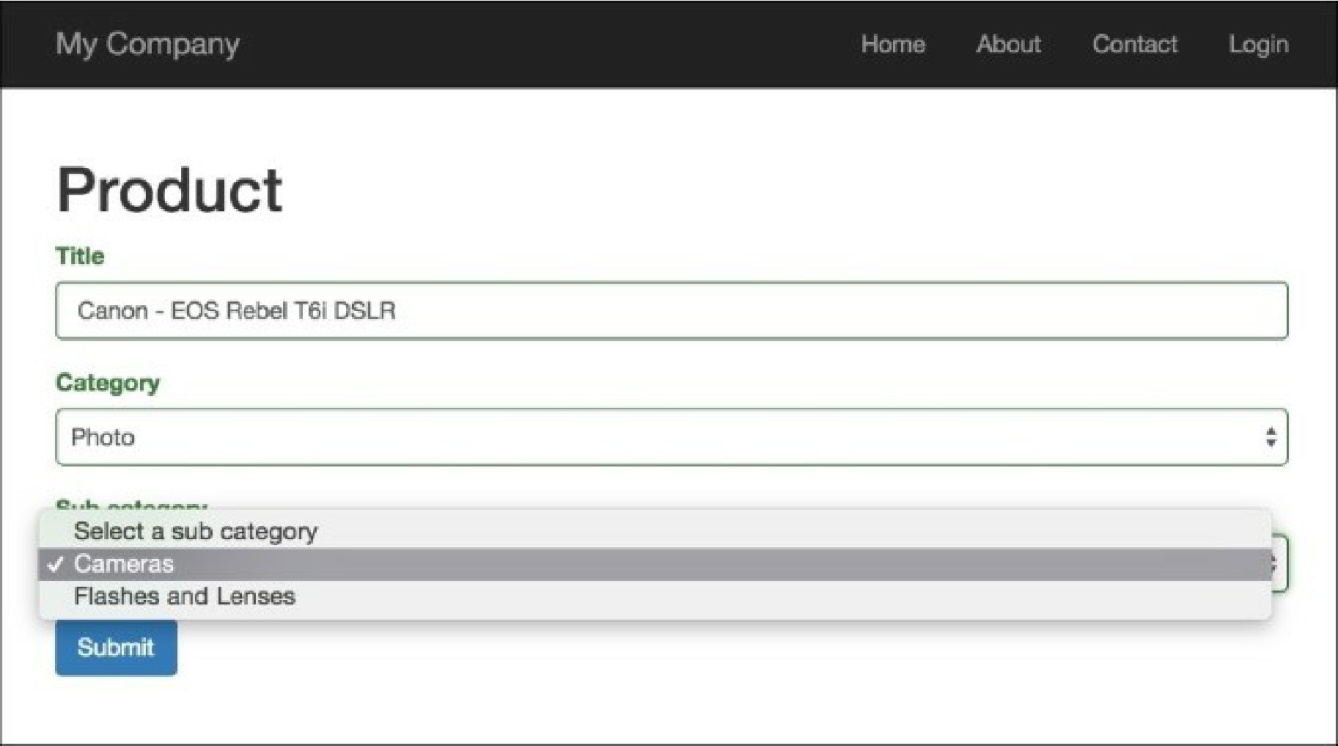
</div>

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

3. Run the dropdown controller by opening index.php?r=dropdown, then add a new product with the  
value Canon - eos Rebel T6i dslr for the title field:



4. As you can see, the Category input has three options. Let’s select the **Photo** option and after that,  
the second input selection will have two further options:



5. That is it. If you select another category, you will get sub-categories of this category.

How it works...

In this example, we have two dependent lists with categories and sub-categories, and one model,  
Category. The main idea is simple: we just bound the JQuery onChange event to the category\_id field in  
our form. Every time a user changes this field, our app sends an AJAX request to the get -sub-  
categories action. This action returns a JSON-formatted list of sub-categories, and then, on the client-  
side, we build a list of options for our sub-categories list.