{

$post = new Post();

$post->title = 'Title';

$post->content\_markdown = 'New \*markdown\* text'; $this->assertTrue($post->save());

$this->assertEquals("<p>New <em>markdown</em> text</p>\n", $post- >content\_html);

}

public function testExistingModelSave()

{

$post = Post::findOne(1);

$post->content\_markdown = 'Other \*markdown\* text'; $this->assertTrue($post->save());

$this->assertEquals("<p>Other <em>markdown</em> text</p>\n", $post- >content\_html);

}

public function fixtures()

{

return [

'posts' => [

'class' => PostFixture::className(),

]

];

}

}

3. Run the unit tests:

codecept run unit MarkdownBehaviorTest Ensure that tests has not passed:

Codeception PHP Testing Framework v2.0.9

Powered by PHPUnit 4.8.27 by Sebastian Bergmann and contributors.

Unit Tests (2)

Trying to test... MarkdownBehaviorTest::testNewModelSave Error

Trying to test. MarkdownBehaviorTest::testExistingModelSave Error

Time: 289 ms, Memory: 16.75MB

4. Now we need to implement behavior, attach it to the model, and make sure the test passes. Create a new directory, behaviors. Under this directory, create a MarkdownBehavior class:

<?php

namespace app\behaviors;

use yii\base\Behavior; use yii\base\Event;

use yii\base\InvalidConfigException; use yii\db\ActiveRecord; use yii\helpers\Markdown;

class MarkdownBehavior extends Behavior {

public $sourceAttribute; public $targetAttribute;

public function init()