if (empty($this->sourceAttribute) || empty($this->targetAttribute)) { throw new InvalidConfigException('Source and target must be set.');

{

}

parent::init();

}

public function events()

{

return [

ActiveRecord::EVENT\_BEFORE\_INSERT => 'onBeforeSave',

ActiveRecord::EVENT\_BEFORE\_UPDATE => 'onBeforeSave',

];

}

public function onBeforeSave(Event $event)

{

if ($this->owner->isAttributeChanged($this->sourceAttribute)) { $this->processContent();

}

}

private function processContent()

{

$model = $this->owner;

$source = $model->{$this->sourceAttribute};

$model->{$this->targetAttribute} = Markdown::process($source);

}

}

5. Let’s attach the behavior to the Post model:

class Post extends ActiveRecord {

public function behaviors()

{

return [

'markdown' => [

'class' => MarkdownBehavior::className(), 'sourceAttribute' => 'content\_markdown', 'targetAttribute' => 'content\_html',

],

];

}

}

6. Run the test and make sure it passes:

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 Ok

Trying to test. MarkdownBehaviorTest::testExistingModelSave Ok

Time: 329 ms, Memory: 17.00MB

7. That’s it. We’ve created a reusable behavior and can use it for all future projects by just connecting it to a model.