Entity

* Cake\ORM\Entity
* Directory: src/Model/Entity

Create entity

* Create from table registry

$user = TableRegistry::get(‘user’)->newEnity();

Properties:

* $\_accessible
* Allow to provider a map of properties
* Example:

protected $\_accessible = [

'title' => true,

'body' => true

];

* In addition to concrete fields there is a special \* field which defines the fallback behavior if a field is not specifically named

protected $\_accessible = [

'title' => true,

'body' => true,

'\*' => false,

];

* Avoiding mass assignment protection

// When creating a **new** entity using the new keyword you can tell it to not protect itself against mass assignment

use App\Model\Entity\Article;

$article = new Article(['id' => 1, 'title' => 'Foo'], ['guard' => false]);

* Modifying the guarded fields at runtime

// You can modify the list of guarded fields at runtime using the accessible method:

// Make user\_id accessible.

$article->accessible('user\_id', true);

// Make title guarded.

$article->accessible('title', false);

Set value for attribute

$user->email = $this->request->getData(‘email’);

Get

Echo $user->email;

Accessors and Mutators

* Custom set get attribute
* Get
* Cake\ORM\Entity::get($field)
* Using: \_get
* Ex: email -> \_getEmail()
* Example:

// in user entity

protected function \_getEmail($email)

{

return ucwords($email);

}

//

echo $user->email;

echo $user->get('email');

* Set
* ~ get
* Ex:

protected function \_setTitle($title)

{

return Text::slug($title);

}

//

$user->title = 'foo'; // slug is set as well

$user->set('title', 'foo'); // slug is set as well

Creating virtual properties

* Table has first\_name and last\_name => create method to get Full name
* Code:

protected function \_getFullName()

{

return $this->\_properties['first\_name'] . ' ' .

$this->\_properties['last\_name'];

}

//

echo $user->full\_name;

Checking if an Entity Has Been Modified

* Cake\ORM\Entity::dirty($field = null, $dirty = null)
* <https://book.cakephp.org/3.0/en/orm/entities.html#checking-if-an-entity-has-been-modified>

validation errors

* Cake\ORM\Entity::errors($field = null, $errors = null)
* Get error:

// get all the errors

$errors = $user->getErrors();

// or, prior to 3.4

$errors = $user->errors();

// get the errors for single field

$errors = $user->getError(‘password’);

// or, prior to 3.4

$errors = $user->error(‘password’);

* Set error:

$user->setError('password', ['Password is required']);

$user->setErrors(['password' => ['Password is required'], 'username' => ['Username is required']]);

// Prior to 3.4.0

$user->errors('password', ['Password is required.']);

Lazy loading associations

* <https://book.cakephp.org/3.0/en/orm/entities.html#lazy-loading-associations>