'name' => 'Raleigh',

'phone' => '1-655-488-3585x699',

'city' => 'Reedstad',

'about' => 'Dolorem quae impedit tempore libero doloribus nobis dicta tempora facere.',

'password' =>

'$2y$13$U7Qte5Y1jVLrx/pnhwdwt.1uXDegGXuNVzEQyUsb65WkBtjyjUuYm',

'auth\_key' => 'uWWJDgy5jNRk6KjqpxS5JuPv0OHearqE',

],

],

Working with your own data types

1. Create your own provider with your custom value generating logic:

<?php

namespace tests\codeception\faker\providers;

use Faker\Provider\Base;

class UserStatus extends Base {

public function userStatus()

{

return $this->randomElement([0, 10, 20, 30]);

}

}

1. Add the provider into the providers list in the /tests/codeception/config/config. php file: return [

'controllerMap' => [

'fixture' => [

'class' => 'yii\faker\FixtureController',

'fixtureDataPath' => '@tests/codeception/fixtures',

'templatePath' => '@tests/codeception/templates',

'namespace' => 'tests\codeception\fixtures',

'providers' => [

'tests\codeception\faker\providers\UserStatus',

],

],

],

// ...

];

1. Add the status field into your fixture template file:

<?php

/\*\*

* @var $faker \Faker\Generator
* @var $index integer \*/

return [

'name' => $faker->firstName,

'status' => $faker->userStatus,

];

1. Regenerate fixtures with the console command:

php tests/codeception/bin/yii fixture/generate users --count=2

1. Check that the generated code in the fixtures/users.php file contains your custom values: