##### Фикстуры.

Для хранения фикстур используется класс **Fixtures** который представляет из себя хранилище данных. Сначала мы сохраняем туда нужные данные (массивы фикстур), а далее можем получить в любом нужном классе. Для удобства создан статический интерфейс методов класса Fixtures, т.е. не нужно получать объект.  
  
Методы класса Fixtures можно посмотреть в файле vendor\codeception\codeception\src\Codeception\Util\Fixtures.php  
Основные:

Fixtures::add('users', ['name' => 'Ivan']); //сохраняем

Fixtures::exists('users'); //проверяем наличие в хранилище

Fixtures::get('users'); //получаем

Согласно текущей документации, фикстуры должны быть заданы в **bootstrap** файле, т.е. единожды и до начала тестирования.   
  
Перед использованием фикстур нужно создать тестовую базу данных. Это делается вручную или используя модуль Db (перезагрузкой дампа).   
Обычно создается чистая тестовая база данных со структурой нужных таблиц, заполнение которых уже происходит с помощью фикстур.  
  
Вы можете использовать в классах **Cest** метод **\_before()** для загрузки фикстур перед началом теста (каждого метода). Метод **\_after** позволяет очистить БД между выполнением тестирующих методов одного класса. Удалять внесенные тестами данные после выполнения каждого теста можно и с помощью модуля Db, если в настройках модуля указать дамп и параметр cleanup: true

dump: 'tests/\_data/test.sql'

cleanup: true

Если быть точным, то данный механиз просто перезапишет базу данных из дампа. И так перед каждым тесто, что занимает сравнительно много времени. Поэтому в моем примере я не буду это использовать.  
  
Но подключить модуль Db для соединения с базой данных нужно в любом случае. Например так в файле **tests\acceptance.suite.yml**:

modules:

enabled:

- Db:

dsn: 'mysql:host=localhost;dbname=test'

user: 'root'

password: ''

В качестве примера создам приемочный тест, работающий с все той же таблицей users.   
  
Для хранения фикстур создаем папку **\_fixtures** в каталоге test. В ней создаем файл фикстур users.php который будет содержать данные для добавления в таблицу users:

<?php

return [

[

'name' => 'testName1',

'email' => 'testName1@gmail.com',

],

[

'name' => 'testName2',

'email' => 'testName2@gmail.com',

]

];

Для удобства воспользуемся файлом предзагрузки **tests\\_bootstrap.php**, создаем если его нет. Определяем константу которая будет содержать путь к папке с фикстурами:

<?php

define("FIXTURES\_DIR", \_\_DIR\_\_ . '/\_fixtures/');

Данная константа может понадобиться для различных типов тестов.  
  
Теперь создаем, файл предзагрузки для приемочных тестов **\_bootstrap.php** в папке **tests\acceptance** с содержимым:

<?php

use Codeception\Util\Fixtures;

/\*

\* Используем класс Fixtures для хранения фикстур

\* FIXTURES\_DIR - константа заданная в главном файле \_bootstrap

\*/

Fixtures::add('users', require(FIXTURES\_DIR . 'users.php'));

Данный код можно указать и в основном файле предзагрузки, созданном выше, если вы планируете использовать указанные фикстуры и для других типов тестов (функциональных и модульных).  
Теперь до начала тестирования у нас будет подключен файл фикстур и сохранен в глобальное хранилище фикстур которое предоставляет класс Fixtures.   
  
Создаем приемочный тест для демонстрации работы с фикстурами:

codecept generate:cest acceptance UserInDb

Добавляем содержимое:

<?php

use Codeception\Util\Fixtures;

class UserInDbCest

{

public function \_before(AcceptanceTester $I)

{

if(Fixtures::exists('users')){

/\*

\* Добавляем строки в таблицу 'users'

\*/

foreach (Fixtures::get('users') as $item) {

$I->haveInDatabase('users', $item);

}

}

}

public function \_after(AcceptanceTester $I)

{

}

public function checkUserInDb(AcceptanceTester $I)

{

$I->wantTo('Проверяем наличие данных в таблице users');

$I->seeInDatabase ('users', array ('name' => 'testName1', 'email' => 'testName1@gmail.com'));

}

}

После запуска этого теста:

codecept run acceptance UserInDbCest

должны получить уведомление об успешном его прохождении. Значит фикстуры отработали.  
  
В данном примере я использовал метод **\_before()** для заполнения таблицы фикстурами (нужными для тестирования данными). Напомню, что данный метод выполняется автоматически перед каждым тестовым методом.   
Строкой

if(Fixtures::exists('users'))

проверяем наличие нужной фикстуры в хранилище по ее ключу. Получаем данные с помощью

Fixtures::get('users')

и далее добавляем массив данных в БД используя метод haveInDatabase():

$I->haveInDatabase('users', $item);

Т.к. массив содержит вложенные массивы каждый из которых представляет собой отдельный набор для добавления в таблицу (отдельная строка) используется цикл.  
  
Видно, что я не использовал метод **\_after()** для очистки таблицы между тестами. Это не потребовалось т.к. я использовал метод **haveInDatabase()** для заполнения данными, а данный метод автоматически удаляет внесенные данные сразу после теста.

Перед каждым запуском нужно заполнять базу тестовыми данными. Для этих вещей можно подключить и настроить модуль Db для импорта дампа, но фреймворк предоставляет работу с тестовыми данными в виде простых PHP-массивов.

Для модульных и интеграционных тестов расширение yii2-codeception поддерживает фикстуры «из коробки». Другое дело – - функциональные и приёмочные. Для поддержки в них миграций можно пойти путём написания собственного плагина для Codeception. Мы, возьмём готовый [FixtureHelper](https://github.com/yiisoft/yii2-app-advanced/blob/master/tests/codeception/common/_support/FixtureHelper.php) из advanced-приложения, поместим его в папку tests/codeception/\_support и поменяем лишь пути и пространство имён:

namespace tests\codeception\\_support;

*// use tests\codeception\fixtures\UserFixture;*use Codeception\Module;

use yii\test\FixtureTrait;

use yii\test\InitDbFixture;

**class** FixtureHelper **extends** Module{

use FixtureTrait {

loadFixtures **as** **protected**;

fixtures **as** **protected**;

globalFixtures **as** **protected**;

unloadFixtures **as** **protected**;

getFixtures **as** **protected**;

getFixture **as** **protected**;

}

**public** **function** \_beforeSuite($settings = [])

{

$this->loadFixtures();

}

**public** **function** \_afterSuite()

{

$this->unloadFixtures();

}

**public** **function** globalFixtures()

{

**return** [

InitDbFixture::className(),

];

}

**public** **function** fixtures()

{

**return** [

*//'user' => [*

*// 'class' => UserFixture::className(),*

*// 'dataFile' => '@tests/codeception/fixtures/data/user.php',*

*//],*

];

}}

Фикстуру UserFixture мы пока закомментировали. Скоро мы её добавим свою.