

FAKE!

CUSTOMIZED FIXTURES WITH **FAKER & ALICE**

by Philipp Rieber

PHP User Group Munich – May 27th 2015

ABOUT://ME

Philipp Rieber

I work at  **session**



SensioLabs
UK

I tweet [@bicpi](https://twitter.com/bicpi)

I'm a technical author

More info at <http://philipp-rieber.net>

FIXTURES

“Fixtures are used to load a controlled set of data into a database. This data can be used for testing or could be the initial data required for the application to run smoothly.”

from the Symfony docs

A USER MODEL

```
class User {  
    public $firstName;  
  
    public $lastName;  
  
    public $email;  
  
    public $city;  
}
```

LISTING THE USERS

First name

?
?
?

Last name

?
?
?

Email

?
?
?

City

?
?
?

USING MANUAL SKILLS

First name	Last name	Email	City
Philipp	Rieber	me@gmail.com	Munich
qweqwe	sdfsdf	fwef@wfe.df	wefwef
first_name	last_name	example@example.org	city

USE SOME AUTOMATION

First name	Last name	Email	City
First 1	Last 1	user-1@example.org	City 1
First 2	Last 2	user-2@example.org	City 2
First 3	Last 3	user-3@example.org	City 3

THE SYMFONY WAY

```
class LoadData extends AbstractFixture
{
    public function load(ObjectManager $manager)
    {
        for($i = 1; $i <= 3; $i++) {
            $user = (new User())
                ->setFirstName("First $i")
                ->setLastName("Last $i");
            ->setEmail("user-$i@example.org");
            ->setCity("City $i");
            $manager->persist($user);
        }
        $manager->flush();
    }
}
```


BUT THERE'S MUCH MORE TO TEST DATA

- Realistic
- Reproducible
- Uniqueness
- Optional data
- Conditions
- Localization

Unique random name

```
$names = ['Klara Fall', 'Rainer Zufall', ...];  
shuffle($names);  
$uniqueName = array_pop($names);
```

Random timezone

```
$timezones = \DateTimeZone::listIdentifiers();  
$timezone = $timezones[array_rand($timezones)];
```

Random german postcode

```
$zip = sprintf('%05d', mt_rand(0, 99999));
```

Random HTML color code

```
$hexColor = sprintf('#%06X', mt_rand(0, 0xFFFFFFFF));
```

Last login – needs to be in the past

```
$someHoursAgo = time() - mt_rand(0, 3600*72);  
$lastLogin = new \DateTime('@'.$someHoursAgo);
```

Random elements from a value set

```
$skills = ['PHP', 'SQL', 'NoSQL' ...];
$users = [];
for ($i = 0; $i < 10; $i++) {
    $userSkills = [];
    $keys = (array) array_rand($skills, mt_rand(1, 3));
    foreach($keys as $key ) {
        $userSkills[] = $skills[$key];
    }
}
```

Optional random text with a probability of 35%

```
$optionalText = mt_rand(0, 100) <= 35  
  ? 'Lorem ipsum dolor sit amet ...'  
  : null;
```


FAKER

A PHP library to create realistic test data

Created by Francois Zaninotto - @francoisz

INSTALLATION

```
$ composer require fzaninotto/Faker
```

Initialize faker generator

```
require_once 'vendor/autoload.php';  
  
$faker = Faker\Factory::create();
```

FAKER MAGIC

```
$faker->title;      // e.g. "Mrs."  
$faker->firstName; // e.g. "Lucy"  
$faker->lastName;  // e.g. "Hessel"  
$faker->email;     // e.g. "inienow@rippin.net"  
$faker->url;       // e.g. "http://reichert.com/sit-autem"  
$faker->ipv4;      // e.g. "116.132.63.32"  
$faker->numberBetween(4, 78) // e.g. "43"
```

FORMATTERS

Generator properties/methods are called *Formatters*

```
$faker->firstName; // e.g. "Lucy"
$faker->firstName(); // e.g. "Lucy"

$faker->firstNameMale; // e.g. "John"

$faker->numberBetween(56, 391); // e.g. 183
$faker->randomElement(['md5', 'sha1', 'sha256']); // e.g. "sha1"

// e.g. DateTime('1982-12-09 13:42:22')
$faker->dateTimeBetween('-63 years', 'now');

// => 'http://loempixel.com/640/480/', random image
$faker->imageUrl(640, 480)
```

PROVIDERS

Formatters are grouped in *Providers*

Person	Address	Internet	...
title	street	ipv6	...
name	postcode	freeEmail	...
suffix	country	companyEmail	...
...

UNIQUE AND OPTIONAL

```
$faker->unique()->email;
```

```
$faker->optional(.35)->text(150);
```

```
$faker->optional(.35, 'n/a')->text;
```

LOCALIZATION

```
$de = Faker\Factory::create('de_DE');
$de->name;           // e.g. "Ingo Barth"
$de->city;           // e.g. "Bremen"
$de->companySuffix; // e.g. "GmbH & Co. KG"

$at = Faker\Factory::create('de_AT');
$at->city;           // e.g. "Gerasdorf bei Wien"

$fr = Faker\Factory::create('fr_FR');
$fr->name;           // e.g. "Hélène Moreau"
$fr->departmentName; // e.g. "Pyrénées-Atlantiques"
```

CUSTOM PROVIDERS & FORMATTERS

```
class MyProvider
{
    public function encodePassword($plain)
    {
        $digest = hash('sha512', $plain, true);
        for ($i = 1; $i < 1000; $i++) {
            $digest = hash('sha512', $plain, true);
        }

        return base64_encode($digest);
    }
}

$faker->addProvider(new MyProvider()); // Register custom provider

$faker->encodePassword('mySecretPassword'); // Usage
```


SEEDING

```
$faker = Faker\Factory::create();  
$faker->seed(1);  
  
$faker->name; // always the same name
```

PERSISTENCE

Persist 50 Doctrine User entities with fake data

```
$populator = new Faker\ORM\Doctrine\Populator($faker, $entityManager);  
$populator->addEntity('User', 50);  
  
$primaryKey = $populator->execute();
```

THE SYMFONY WAY

```
class LoadData extends AbstractFixture implements ContainerAwareInterface
{
    public function load(ObjectManager $manager)
    {
        $faker = $this->container->get('faker');
        for($i = 1; $i <= 3; $i++) {
            $user = (new User())
                ->setFirstName($faker->firstName)
                ->setLastName($faker->lastName);
            ->setEmail($faker->email);
            ->setCity($faker->city);
            $manager->persist($user);
        }
        $manager->flush();
    }
}
```

ALICE

A PHP library to generate and organize fixtures

Created by the Nelmio company

INSTALLATION

```
$ composer require nelmio/alice
```

```
require_once 'vendor/autoload.php';  
  
$loader = new Nelmio\Alice\Loader\Yaml();  
  
$objects = $loader->load(__DIR__.'/fixtures.yml');
```

MAPPING DATA TO OBJECTS

```
# fixtures.yml
Acme\Entity\User:
  user_max:
    firstName: Max
    lastName: Mustermann
    email: max@mustermann.de
    city: Munich
  user_erika:
    firstName: Erika
    lastName: Musterfrau
    email: erika@musterfrau.de
    city: Berlin

# Acme\Entity\Team:
# ...
```

OBJECT RANGE

```
Acme\Entity\User:  
  user_{1..50}:  
    name: John Doe  
    email: john@doe.net
```

DYNAMIC DATA

```
Acme\Entity\User:  
  user_{1..50}:  
    name: John Doe <current()>  
    email: john-<current()>@doe.net
```


FAKER INTEGRATION

```
Acme\Entity\User:  
  user_{1..50}:  
    name: <name()>  
    email: <email()>
```

UNIQUE FIELDS

```
Acme\Entity\User:  
  user_{1..50}:  
    name: <name()>  
    email (unique): <email()>
```

OPTIONAL FIELDS

```
Acme\Entity\User:  
  user_{1..50}:  
    name: <name()>  
    bio: 35%? <text()> : n/a
```

LOCALIZATION

```
Acme\Entity\User:  
  user_{1..50}:  
    name: <de_DE:name()>  
    email: <fr_FR:email()>
```

OBJECT REFERENCES

```
Acme\Entity\User:
  user_{1..50}:
    name: <name()>
    email: <email()>

Acme\Entity\Team:
  team:
    members: [@user_11, @user_38, @user_41]
```

OBJECT REFERENCES

```
Acme\Entity\Team:  
  team:  
    users: [@user_*, @user_*, @user_*]
```

OBJECT REFERENCES

```
Acme\Entity\Team:  
  team:  
    users: 3x @user_*
```

OBJECT REFERENCES

```
Acme\Entity\Team:  
  team:  
    users: <numberBetween(1, 3)>x @user_*
```


PERSISTENCE

```
$loader = new \Nelmio\Alice\Fixtures\Loader();  
$objects = $loader->load(__DIR__.'/fixtures.yml');  
  
$persister = new \Nelmio\Alice\Persister\Doctrine($objectManager);  
$persister->persist($objects);
```

Shortcut:

```
$objects = Nelmio\Alice\Fixtures::load(  
    __DIR__.'/fixtures.yml', $objectManager  
);
```

ADVANCED

And there's much more ...

- References: @self, @user_2->email
- Templates
- Inheritance
- Parameters
- File Includes
- Variables
- Value Objects
- Processors
- ...

ALICEBUNDLE

Faker & Alice integration with Symfony and Doctrine

INSTALLATION & ACTIVATION

```
$ composer require doctrine/doctrine-fixtures-bundle
$ composer require hautelook/alice-bundle
```

```
class AppKernel extends Kernel
{
    public function registerBundles()
    {
        $bundles = array(
            // ...
        );

        if (in_array($this->getEnvironment(), array('dev', 'test'))) {
            $bundles[] = new Doctrine\Bundle\FixturesBundle\DoctrineFixturesBu
            $bundles[] = new Hautelook\AliceBundle\HautelookAliceBundle();
        }
    }
}
```

CONFIGURATION

```
# config.yml
hautelook_alice:
  locale: de_DE
  seed: 1
```

USAGE

```
namespace App\AppBundle\DataFixtures\ORM;

use Hautelook\AliceBundle\Alice\DataFixtureLoader;
use Nelmio\Alice\Fixtures;

class DataLoader extends DataFixtureLoader
{
    protected function getFixtures()
    {
        return [__DIR__ . '/fixtures.yml'];
    }
}
```

```
$ app/console doctrine:fixtures:load
```

CUSTOM FORMATTERS

```
use \Faker\Provider\Base as FakerBase;

class DataLoader extends DataFixtureLoader
{
    protected function getFixtures()
    {
        return [__DIR__ . '/fixtures.yml'];
    }

    public function encodePassword($plain)
    {
        // ...
    }
}
```

```
# fixtures.yml
User:
    password: <encodePassword('mySecretPassword')>
```

SUMMARY

When using Symfony and the AliceBundle you have

- a little loader class with almost nothing in it
- straightforward YAML files with test data

RESOURCES

- [Faker library](#)
- [Alice library](#)
- [AliceBundle](#)
- [KnpUniversity Screencast](#) – \$3



Slides

<http://philipp-rieber.net#talks>

Contact

@bicpi