

CYDEO

JavaFaker



Agenda :

- Understand what JavaFaker is.
- Understand why/when do we need it.
- Understand how to implement it.

Why do we need JavaFaker?

- Let's understand the challenge:
- If you are working on an application where you must enter lots of test data, it might just extra work to generate all that data.
- There is a library that helps with this situation.

What is JavaFaker?

- JavaFaker is an external dependency with a set of classes and methods
- JavaFaker library helps us automate generating random data.
- We will use the data as?
- Test Data.

How do we implement JavaFaker dependency?

- Same as how we implement any other dependency in a Maven project.
 1. Find the dependency.
 2. Copy it from the source.
 3. Paste it into our pom.xml file.
 4. Refresh our Maven project to have it installed.

Let's observe what options we have in GitHub repository.

<https://github.com/DiUS/java-faker>

How do we generate test data using JavaFaker?

1. Create JavaFaker object

```
Faker faker = new Faker();
```

2. Start using the object to reach ready methods

```
String name = faker.name().fullName(); // Miss Jane Watson
```

```
String firstName = faker.name().firstName(); // Carol  
String lastName = faker.name().lastName(); // Denver
```

```
String streetAddress = faker.address().streetAddress();  
// 70048 Dee Road Suite 415
```

```
String cityName = faker.address().cityName();  
//Stephanberg
```

```
String zipCode = faker.address().zipCode();  
//45535
```

Let's implement it together

- Go to google
- Search for: javafaker github
- Copy the dependency
- Paste it into pom.xml
- Refresh project.