

# ToiletPaper #5

## RandomBeans

Von Gernot Pointner

### ✘ Problem

Setup-Code für Tests mit **komplexen POJOs** zu schreiben ist anstrengend und **fehleranfällig**.

Dies führt oft dazu, dass **Test-Setup-Code geteilt** und in mehreren Einheitstests eingesetzt wird, was wiederum zu **einer engen Verbindung** zwischen Testklassen führt. Weiterhin ist es nahezu **unmöglich, alle** möglichen POJO-Kombinationen zu erstellen und alle **Ausnahmefälle** abzudecken.

### ✓ Lösung

Nutze die **RandomBeans**-Library um zufällige Beispiele für deine Testfälle zu generieren.

Sie bietet **sprechende API** zur Spezifizierung von Wertebereichen, festen Werten oder einer Gruppe von Werten für die erstellten Testinstanzen. Sie ermöglicht auch die Generierung von **komplexeren POJOs** (z.B. rekursive Objektgraphen). Mithilfe von RandomBeans können **unterschiedliche Testinstanzen für jeden Testlauf generiert werden**. Falls ein Test nicht erfolgreich ist, ist es möglich, das exakt gleiche TestszENARIO mit denselben Anfangswerten durchzuführen.

Einige **mögliche Anwendungsfälle** sind:

- Auffüllen einer Testdatenbank mit zufälligen Einträgen
- Generierung von zufälligen Anfragen oder Antworten für deine RESTful API
- Generierung von zufälligem Input für deine Validatoren

### ➔ Beispiel

```
// example User POJO - setters/getters omitted for the sake of brevity
public class User {
    public String name;
    public List<String> hobbies;
    public LocalDate birthday;
    public URL website;
}
// configure a EnhancedRandom for creating User instances
// * generate UTF-8 String with 10 to 15 characters
// * created LocalDate are within a range from 50 years ago till now
// * collections should contain from 1 up to 20 elements
// * the field "website" of our User should have a valid URL
EnhancedRandom er = EnhancedRandomBuilder.aNewEnhancedRandomBuilder()
    .charset(Charset.forName("UTF-8"))
    .stringLengthRange(10, 15)
    .dateRange(now().minusYears(50), now())
    .collectionSizeRange(1, 20)
    .randomize(field().named("website").ofType(URL.class).inClass(User.class).get(),
        UrlRandomizer.aNewUrlRandomizer())
    .build();

// create test object using our previously configured EnhancedRandom
User user = er.nextObject(User.class);
```

### + Links

- <https://github.com/benas/random-beans>