

Tutoriel de 10 minutes

Dans ce didacticiel rapide, vous apprendrez à :

- Installer le concombre
- Écrivez votre premier scénario en utilisant la syntaxe Gherkin
- Écrivez votre définition de première étape en Java
- Exécuter le concombre
- Apprenez le flux de travail de base du développement piloté par le comportement (BDD)

Nous utiliserons Cucumber pour développer une petite bibliothèque capable de déterminer si nous sommes déjà vendredi.

Veuillez noter que ce tutoriel suppose que vous disposez d'un :

- Compréhension de base du langage de programmation Java
 - Une certaine expérience de l'utilisation d'un terminal
 - Une certaine expérience de l'utilisation d'un éditeur de texte

Avant de commencer, vous aurez besoin des éléments suivants :

- [JavaSE](#)
- Un outil de construction. Vous pouvez choisir entre :
 - [Maven](#) - version 3.3.1 ou supérieure
- [IntelliJ IDEA](#) (qui sera utilisé dans ce tutoriel)
 - [Plugin IntelliJ IDEA Concombre pour Java](#)

Créer un projet Concombre vide

Nous allons commencer par créer un nouveau répertoire de projet avec le **cucumber-archetype** plugin Maven. Ouvrez un terminal, accédez au répertoire dans lequel vous souhaitez créer votre projet et exécutez la commande suivante :

```
mvn archetype:generate "-DarchetypeGroupId=io.cucumber" "-DarchetypeArtifactId=cucumber-archetype" "-DarchetypeVersion=7.18.0" "-DgroupId=hello cucumber" "-DartifactId=hello cucumber" "-Dpackage=hello cucumber" "-Dversion=1.0.0-SNAPSHOT" "-DinteractiveMode=false"
```

Vous devriez obtenir quelque chose comme le résultat suivant :

```
[INFO] Project created from Archetype in dir: <directory where you created the project>/cucumber

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----
```

Accédez au répertoire qui vient d'être créé en exécutant la commande suivante :

```
cd hellocucumber
```

Ouvrez le projet dans IntelliJ IDEA :

- **Fichier -> Ouvrir... -> (Sélectionnez le pom.xml)**
- Sélectionnez **Ouvrir en tant que projet**

Vous avez maintenant un petit projet avec Cucumber installé.

Vérifier l'installation de Cucumber

Pour nous assurer que tout fonctionne correctement, exécutons Cucumber.

```
mvn test
```

Vous devriez voir quelque chose comme ce qui suit :

```
-----
T E S T S
-----

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

La sortie de Cucumber nous indique qu'il n'a rien trouvé à exécuter.

Écrire un scénario

Lorsque nous effectuons un développement piloté par le comportement avec Cucumber, nous utilisons *des exemples concrets* pour spécifier *ce que* nous voulons que le logiciel fasse. Les scénarios sont écrits *avant* le code de production. Ils commencent leur vie en tant que *spécification exécutable*. À mesure que le code de production émerge, les scénarios jouent le rôle de *documentation vivante* et de *tests automatisés*.

Exemple de mappage

Essayez d'organiser un atelier [de cartographie d'exemples](#) dans votre équipe pour concevoir des exemples ensemble.

Dans Cucumber, un exemple s'appelle un **scénario**. Les scénarios sont définis dans **.feature** des fichiers stockés dans le répertoire (ou un sous-répertoire) **src/test/resources/hellocucumber**

Un exemple concret serait que *dimanche n'est pas vendredi*.

Créez un fichier vide appelé avec le contenu

suivant : **src/test/resources/hellocucumber/is_it_friday_yet.feature**

```
Feature: Is it Friday yet?
```

```
    Everybody wants to know when it's Friday
```

```
Scenario: Sunday isn't Friday
```

```
    Given today is Sunday
```

```
    When I ask whether it's Friday yet
```

```
    Then I should be told "Nope"
```

La première ligne de ce fichier commence par le mot-clé **Feature:** suivi d'un nom. C'est une bonne idée d'utiliser un nom similaire au nom du fichier.

La deuxième ligne est une brève description de la fonctionnalité. Cucumber n'exécute pas cette ligne car il s'agit de documentation.

La quatrième ligne **Scenario: Sunday is not Friday** est un **scénario**, qui est un *exemple concret* illustrant le comportement du logiciel.

Les trois dernières lignes commençant par **Given**, **When** et **Then** constituent les **étapes** de notre scénario. C'est ce que Cucumber va exécuter.

Voir le scénario signalé comme non défini

Maintenant que nous avons un scénario, nous pouvons demander à Cucumber de l'exécuter.

```
mvn test
```

Le concombre nous dit que nous avons un **undefined** scénario et trois **undefined** étapes. Cela suggère également quelques extraits de code que nous pouvons utiliser pour *définir* ces étapes :

```
-----
T E S T S
-----
Running hellocucumber.RunCucumberTest

Scenario: Sunday isn't Friday          # hellocucumber/is_it_friday_yet.feature:4
  Given today is Sunday
  When I ask whether it's Friday yet
  Then I should be told "Nope"

Share your Cucumber Report with your team at https://reports.cucumber.io
Activate publishing with one of the following:

src/test/resources/cucumber.properties:    cucumber.publish.enabled=true
src/test/resources/junit-platform.properties:  cucumber.publish.enabled=true
Environment variable:                      CUCUMBER_PUBLISH_ENABLED=true
JUnit:                                      @CucumberOptions(publish = true)

More information at https://cucumber.io/docs/cucumber/environment-variables/
```

Disable this message with one of the following:

src/test/resources/cucumber.properties:	cucumber.publish.quiet=true
src/test/resources/junit-platform.properties:	cucumber.publish.quiet=true

```
[ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.15 s <<< FAILURE! - in hellocucumber.RunCucumberTest
[ERROR] Is it Friday yet?.Sunday isn't Friday Time elapsed: 0.062 s <<< ERROR!
io.cucumber.junit.platform.engine.UndefinedStepException:
The step 'today is Sunday' and 2 other step(s) are undefined.
You can implement these steps using the snippet(s) below:
```

```
@Given("today is Sunday")
public void today_is_sunday() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
@When("I ask whether it's Friday yet")
public void i_ask_whether_it_s_friday_yet() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
@Then("I should be told {string}")
public void i_should_be_told(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
```

Copiez chacun des trois extraits pour les étapes non définies et collez-les dans `src/test/java/hellocucumber/StepDefinitions.java`

Voir le scénario signalé comme en attente

Exécutez à nouveau Cucumber. Cette fois, le résultat est un peu différent :

```
-----
T E S T S
-----
```

```
Running hellocucumber.RunCucumberTest
```

```
Feature: Is it Friday yet?
```

```
    Everybody wants to know when it'sFriday
```

```
    Scenario: Sunday isn't Friday # hellocucumber/is_it_friday_yet.feature:4
```

```
        Given today is Sunday # Stepdefs.today_is_Sunday()
```

```
            io.cucumber.java.PendingException: TODO: implement me
```

```
                at hellocucumber.Stepdefs.today_is_Sunday(StepDefinitions.java:14)
```

```
                at ?.today is Sunday(classpath:hellocucumber/is_it_friday_yet.feature:5)
```

```
        When I ask whether it's Friday yet # Stepdefs.i_ask_whether_it_s_Friday_yet()
```

```
        Then I should be told "Nope" # Stepdefs.i_should_be_told(String)
```

```
Pending scenarios:
```

```
hellocucumber/is_it_friday_yet.feature:4 # Sunday isn't Friday
```

```
1 Scenarios (1 pending)
```

```
3 Steps (2 skipped, 1 pending)
```

0m0.188s

```
io.cucumber.java.PendingException: TODO: implement me
    at hellocucumber.Stepdefs.today_is_Sunday(StepDefinitions.java:13)
    at ?.today is Sunday(classpath:hellocucumber/is_it_friday_yet.feature:5)
```

Cucumber a trouvé nos définitions d'étapes et les a exécutées. Ils sont actuellement marqués comme *en attente*, ce qui signifie que nous devons leur faire faire quelque chose d'utile.

Voir le scénario signalé comme échec

L'étape suivante consiste à faire ce que les commentaires dans les définitions d'étape nous disent de faire :

Écrivez ici du code qui transforme la phrase ci-dessus en actions concrètes

Essayez d'utiliser les mêmes mots dans le code que dans les étapes.

Langue omniprésente

Si les mots de vos étapes proviennent de conversations au cours d'une session [d'exemple de mappage](#), vous créez un [langage omniprésent](#), qui, selon nous, est un excellent moyen de rendre votre code de production et vos tests plus compréhensibles et plus faciles à maintenir.

Remplacez votre code de définition d'étape par ceci :

```
package hellocucumber;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
import static org.junit.jupiter.api.Assertions.*;

class IsItFriday {
    static String isItFriday(String today) {
        return null;
    }
}

public class Stepdefs {
    private String today;
    private String actualAnswer;

    @Given("today is Sunday")
    public void today_is_Sunday() {
        today = "Sunday";
    }

    @When("I ask whether it's Friday yet")
    public void i_ask_whether_it_s_Friday_yet() {
        actualAnswer = IsItFriday.isItFriday(today);
    }

    @Then("I should be told {string}")
```

```

    public void i_should_be_told(String expectedAnswer) {
        assertEquals(expectedAnswer, actualAnswer);
    }
}

```

Exécutez à nouveau Cucumber :

```

-----
T E S T S
-----
Running hellocucumber.RunCucumberTest
Feature: Is it Friday yet?
  Everybody wants to know when it's Friday

  Scenario: Sunday isn't Friday          # hellocucumber/is_it_friday_yet.feature:4
    Given today is Sunday                # Stepdefs.today_is_Sunday()
    When I ask whether it's Friday yet   # Stepdefs.i_ask_whether_it_s_Friday_yet()
    Then I should be told "Nope"         # Stepdefs.i_should_be_told(String)
      java.lang.AssertionError: expected:<Nope> but was:<null>
        at org.junit.Assert.fail(Assert.java:88)
        at org.junit.Assert.failNotEquals(Assert.java:834)
        at org.junit.Assert.assertEquals(Assert.java:118)
        at org.junit.Assert.assertEquals(Assert.java:144)
        at hellocucumber.Stepdefs.i_should_be_told(StepDefinitions.java:31)
        at ?.I should be told "Nope"(classpath:hellocucumber/is_it_friday_yet.feature:7)

Failed scenarios:
hellocucumber/is_it_friday_yet.feature:4 # Sunday isn't Friday

1 Scenarios (1 failed)
3 Steps (1 failed, 2 passed)
0m0.404s

```

C'est un progrès ! Les deux premières étapes réussissent, mais la dernière échoue.

Voir le scénario signalé comme réussi

Faisons le minimum nécessaire pour que le scénario passe. Dans ce cas, cela signifie que notre méthode renvoie **Nope**

```

static String isItFriday(String today) {
    return "Nope";
}

```

Exécutez à nouveau Cucumber :

```

-----
T E S T S
-----
Running hellocucumber.RunCucumberTest
Feature: Is it Friday yet?
  Everybody wants to know when it's Friday

  Scenario: Sunday isn't Friday          # hellocucumber/is_it_friday_yet.feature:4

```

```
Given today is Sunday           # Stepdefs.today_is_Sunday()
When I ask whether it's Friday yet # Stepdefs.i_ask_whether_it_s_Friday_yet()
Then I should be told "Nope"      # Stepdefs.i_should_be_told(String)
```

```
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0.255s
```

Toutes nos félicitations! Vous avez votre premier scénario de concombre vert.

Ajouter un autre test échoué

La prochaine chose à vérifier serait que nous obtenions également le bon résultat *vendredi*.

Mettez à jour le `is_it_friday_yet.feature` fichier :

```
Feature: Is it Friday yet?
```

```
    Everybody wants to know when it's Friday
```

```
    Scenario: Sunday isn't Friday
```

```
        Given today is Sunday
```

```
        When I ask whether it's Friday yet
```

```
        Then I should be told "Nope"
```

```
    Scenario: Friday is Friday
```

```
        Given today is Friday
```

```
        When I ask whether it's Friday yet
```

```
        Then I should be told "TGIF"
```

Nous devons ajouter une définition d'étape à définir `today` sur « Vendredi » :

```
@Given("today is Friday")
public void today_is_Friday() {
    today = "Friday";
}
```

Lorsque nous exécuterons ce test, il échouera.

```
-----
T E S T S
-----
```

```
Running hellocucumber.RunCucumberTest
```

```
Feature: Is it Friday yet?
```

```
    Everybody wants to know when it's Friday
```

```
    Scenario: Sunday isn't Friday           # hellocucumber/is_it_friday_yet.feature:4
```

```

    Given today is Sunday                # Stepdefs.today_is_Sunday()
    When I ask whether it's Friday yet    # Stepdefs.i_ask_whether_it_s_Friday_yet()
    Then I should be told "Nope"          # Stepdefs.i_should_be_told(String)

Scenario: Friday is Friday                # hellocucumber/is_it_friday_yet.feature:9
    Given today is Friday                  # Stepdefs.today_is_Friday()
    When I ask whether it's Friday yet    # Stepdefs.i_ask_whether_it_s_Friday_yet()
    Then I should be told "TGIF"          # Stepdefs.i_should_be_told(String)
    org.junit.ComparisonFailure: expected:<[TGIF]> but was:<[Nope]>
      at org.junit.Assert.assertEquals(Assert.java:115)
      at org.junit.Assert.assertEquals(Assert.java:144)
      at hellocucumber.Stepdefs.i_should_be_told(StepDefinitions.java:36)
      at ?.I should be told "TGIF"(classpath:hellocucumber/is_it_friday_yet.feature:12)

```

Failed scenarios:

hellocucumber/is_it_friday_yet.feature:9 # Friday is Friday

2 Scenarios (1 failed, 1 passed)

6 Steps (1 failed, 5 passed)

0m0.085s

```

org.junit.ComparisonFailure: expected:<[TGIF]> but was:<[Nope]>
  at org.junit.Assert.assertEquals(Assert.java:115)
  at org.junit.Assert.assertEquals(Assert.java:144)
  at hellocucumber.Stepdefs.i_should_be_told(StepDefinitions.java:36)
  at ?.I should be told "TGIF"(classpath:hellocucumber/is_it_friday_yet.feature:12)

```

C'est parce que nous n'avons pas encore implémenté la logique ! Faisons-le ensuite.

Fais-le passer

today Nous devrions mettre à jour notre déclaration pour évaluer réellement si est égal ou non à "Friday".

```

static String isItFriday(String today) {
    return "Friday".equals(today) ? "TGIF" : "Nope";
}

```

Exécutez à nouveau Cucumber :

```

-----
T E S T S
-----
Running hellocucumber.RunCucumberTest
Feature: Is it Friday yet?
  Everybody wants to know when it's Friday

Scenario: Sunday isn't Friday                # hellocucumber/is_it_friday_yet.feature:4
  Given today is Sunday                      # Stepdefs.today_is_Sunday()
  When I ask whether it's Friday yet          # Stepdefs.i_ask_whether_it_s_Friday_yet()
  Then I should be told "Nope"                # Stepdefs.i_should_be_told(String)

Scenario: Friday is Friday                    # hellocucumber/is_it_friday_yet.feature:9
  Given today is Friday                      # Stepdefs.today_is_Friday()
  When I ask whether it's Friday yet          # Stepdefs.i_ask_whether_it_s_Friday_yet()

```



```
Then I should be told "TGIF" # Stepdefs.i_should_be_told(String)
```

```
2 Scenarios (2 passed)
6 Steps (6 passed)
0m0.255s
```

Utiliser des variables et des exemples

Ainsi, nous savons tous qu'il y a plus de jours dans la semaine que le dimanche et le vendredi. Mettons à jour notre scénario pour utiliser des variables et évaluer plus de possibilités. Nous utiliserons des variables et des exemples pour évaluer vendredi, dimanche et tout le reste !

Mettez à jour le `is_it_friday_yet.feature` fichier. Remarquez comment nous passons de `Scenario` à `Scenario Outline` quand nous commençons à utiliser plusieurs `Examples`.

Feature: Is it Friday yet?

Everybody wants to know when it's Friday

Scenario Outline: Today is or is not Friday

Given today is "<day>"

When I ask whether it's Friday yet

Then I should be told "<answer>"

Examples:

day	answer
Friday	TGIF
Sunday	Nope
anything else!	Nope

Nous devons remplacer les définitions d'étape pour `today is Sunday` et `today is Friday` par une définition d'étape qui prend la valeur de `<day>` comme chaîne. Mettez à jour le fichier comme suit : `StepDefinitions.java`

```
package hellocucumber;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
import static org.junit.jupiter.api.Assertions.*;

class IsItFriday {
    static String isItFriday(String today) {
        return "Friday".equals(today) ? "TGIF" : "Nope";
    }
}
```

```

public class Stepdefs {
    private String today;
    private String actualAnswer;

    @Given("today is {string}")
    public void today_is(String today) {
        this.today = today;
    }

    @When("I ask whether it's Friday yet")
    public void i_ask_whether_it_s_Friday_yet() {
        actualAnswer = IsItFriday.isItFriday(today);
    }

    @Then("I should be told {string}")
    public void i_should_be_told(String expectedAnswer) {
        assertEquals(expectedAnswer, actualAnswer);
    }
}

```

Exécutez à nouveau Cucumber :

```

-----
T E S T S
-----
Running hellocucumber.RunCucumberTest
Feature: Is it Friday yet?
    Everybody wants to know when it's Friday

    Scenario Outline: Today is or is not Friday # hellocucumber/is_it_friday_yet.feature:4
        Given today is "<day>"
        When I ask whether it's Friday yet
        Then I should be told "<answer>"

        Examples:

        Scenario Outline: Today is or is not Friday # hellocucumber/is_it_friday_yet.feature:11
            Given today is "Friday" # Stepdefs.today_is(String)
            When I ask whether it's Friday yet # Stepdefs.i_ask_whether_it_s_Friday_yet()
            Then I should be told "TGIF" # Stepdefs.i_should_be_told(String)

        Scenario Outline: Today is or is not Friday # hellocucumber/is_it_friday_yet.feature:12
            Given today is "Sunday" # Stepdefs.today_is(String)
            When I ask whether it's Friday yet # Stepdefs.i_ask_whether_it_s_Friday_yet()
            Then I should be told "Nope" # Stepdefs.i_should_be_told(String)

        Scenario Outline: Today is or is not Friday # hellocucumber/is_it_friday_yet.feature:13
            Given today is "anything else!" # Stepdefs.today_is(String)
            When I ask whether it's Friday yet # Stepdefs.i_ask_whether_it_s_Friday_yet()
            Then I should be told "Nope" # Stepdefs.i_should_be_told(String)

3 Scenarios (3 passed)
9 Steps (9 passed)
0m0.255s

```

Refactorisation

Maintenant que nous avons du code fonctionnel, nous devrions procéder à une refactorisation :

- Nous devrions déplacer la `isItFriday` méthode du code de test vers le code de production.
- Nous pourrions à un moment donné extraire des méthodes d'assistance de notre définition d'étape, pour les méthodes que nous utilisons à plusieurs endroits.

Résumé

Dans ce bref tutoriel, vous avez vu comment installer Cucumber, comment suivre le processus BDD pour développer une méthode et comment utiliser cette méthode pour évaluer plusieurs scénarios !