R4.02 – TP 1

TESTS JUNIT SIMPLES ET PARAMETRES

Table des matières

Objectif

L'objectif est de réaliser des tests unitaires simples uniquement avec JUnit 5.

A lire avec attention

Ne pas oublier de démarrer Docker.

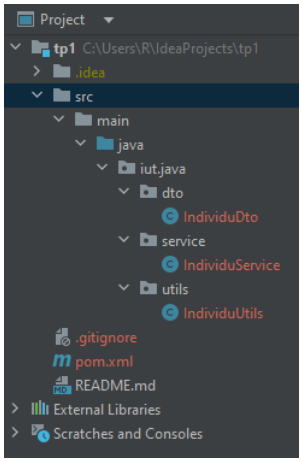
Les énoncés sont à lire avec attention : le respect ou non-respect des indications sera pris en compte dans la notation

Les principales informations pour réaliser le TP sont dans le support de cours. Il n'est pas exclu de compléter/confirmer certaines de ces informations grâce à Internet.

Travaux Pratiques

Création du dépôt et ajout des sources

1. Créer un dépôt « `TP1` » dans le groupe précédemment créé et le cloner.
2. Ajouter les sources contenues dans « `TP1.zip` » et les pousser sur GitLab

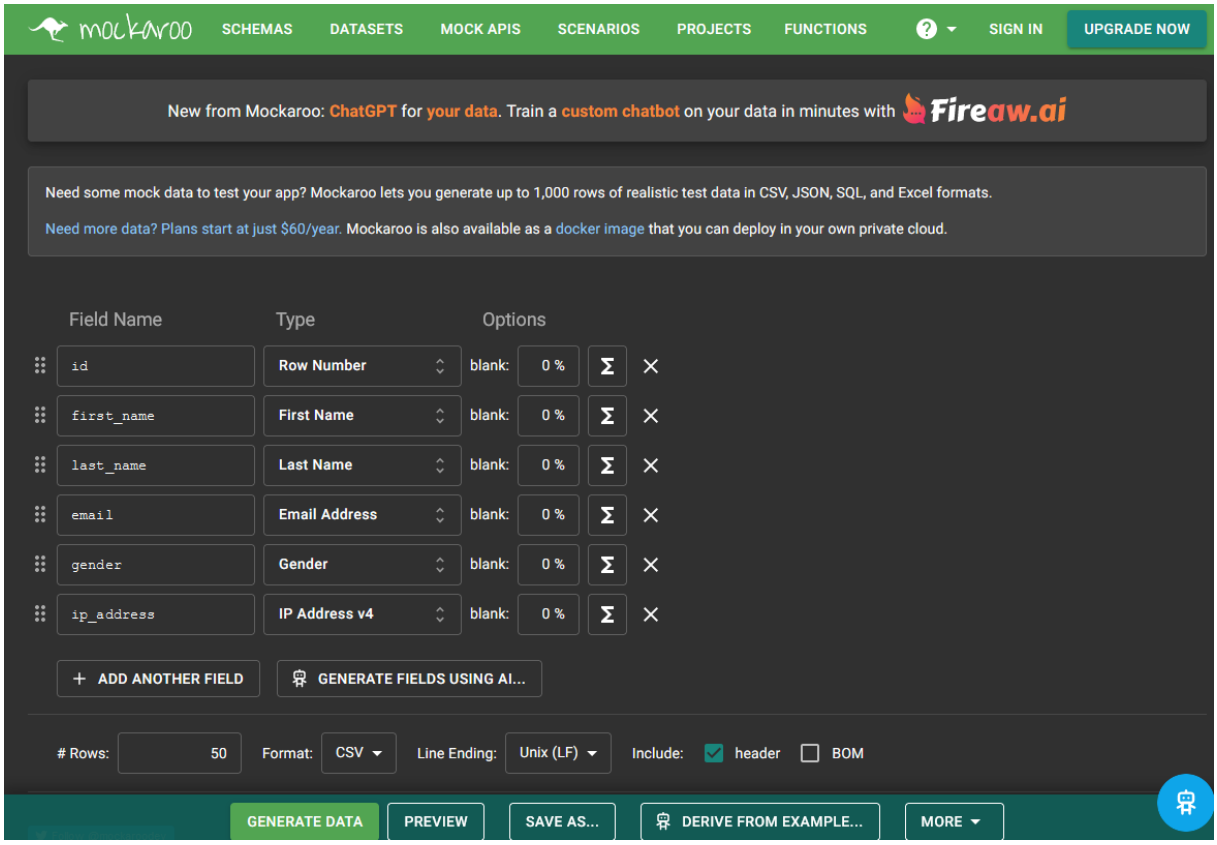


Ajout de la chaine CI/CD

1. Ajouter une chaîne d'intégration continue pour exécuter les tests dans le GitLab Runner sous Docker grâce à Maven.
2. Commiter et vérifier l'exécution de la chaine CI/CD.
3. Si création de la chaine CI/CD depuis GitLab, récupérer les sources depuis GitLab pour ne pas avoir de conflits

Ajout d'un jeu de données aléatoires individuelles

1. Aller sur le site Mockaroo pour générer un fichier CSV avec 50 valeurs



2. Placer le fichier généré dans le dossier « `src/test/resources` ».

```
id,first_name,last_name,email,gender,ip_address
1,Barby,Wedon,bwedon0@woothemes.com,Female,229.34.192.162
2,Matthiew,Boardman,mboardman1@washingtonpost.com,Male,148.184.146.202
3,Shannon,Peasnone,speasnone2@nature.com,Male,76.239.16.91
4,Hans,Foskin,hfoskin3@bbc.co.uk,Polygender,126.209.138.182
5,Cassie,Fasset,cfasset4@so-net.ne.jp,Female,52.114.19.143
6,Ermanno,Ameer-Beg,eameerbeg5@miibeian.gov.cn,Male,245.40.229.232
```

3. Nommer ce fichier avec votre nom/prénom et l'extension CSV.
4. Commiter et vérifier l'exécution de la chaine CI/CD

Test simple de conversion de texte en objet

1. Créer une classe « `IndividuUtilsTest` » dans le package « `iut.java.tests.unit` » du dossier « `src/test/java` »
2. Dans cette classe, créer une méthode de test nommée « `testGetIndividu` » sur la méthode « `getIndividu` » de la classe « `IndividuUtils` »
3. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
4. Cette méthode devra être commentée (pas besoin de faire un roman)
5. Il est possible (voire recommandé) d'utiliser une ligne du fichier CSV généré par Mockaroo pour faire le test.
6. Toutes les valeurs de l'individu doivent être vérifier.
7. Commiter et vérifier l'exécution de la chaine CI/CD

Test simple de conversion d'objet en texte

1. Toujours dans cette classe de test, créer une méthode de test nommée « `testGetLine` » sur la méthode « `getLine` » de « `IndividuUtils` »
2. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
3. Cette méthode devra être commentée (pas besoin de faire un roman)
4. Pour le jeu d'essai (nom et prénom, …), prendre par exemple un personnage ou un artiste (ou soi-même).
5. Commiter et vérifier l'exécution de la chaine CI/CD

Test paramétré de conversion de texte en objet

1. Toujours dans cette classe de test, créer une méthode de test paramétré nommée « `testGetIndividuParam` » sur la méthode « `getIndividu` » de la classe « `IndividuUtils` »
2. La source des paramètres devra être une source CSV (attention au délimiteur).
3. Cette méthode doit permettre de tester la conversion de 3 individus.
4. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
5. Cette méthode devra être commentée (pas besoin de faire un roman)
6. Il est possible (voire recommandé) d'utiliser d'autres lignes du fichier CSV généré par Mockaroo pour faire le test.
7. Toutes les valeurs de l'individu doivent être vérifier.
8. Commiter et vérifier l'exécution de la chaine CI/CD

Test paramétré de conversion d'objet en texte

1. Toujours dans cette classe de test, créer une méthode de test paramétré nommée « `testGetLineParam` » sur la méthode « `getLine` » de « `IndividuUtils` »
2. Cette méthode doit permettre de tester la conversion de 3 individus.
3. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
4. Cette méthode devra être commentée (pas besoin de faire un roman)
5. Il est possible (voire recommandé) d'utiliser d'autres lignes du fichier CSV généré par Mockaroo pour faire le test.
6. Commiter et vérifier l'exécution de la chaine CI/CD

Test du nombre d'individus chargés depuis le fichier CSV

1. Créer une classe « `IndividuServiceTest` » dans le package « `iut.java.tests.integration` » du dossier « `src/test/java` »
2. Dans cette classe, créer une méthode de test nommée « `testCountLoadedIndividus` » sur la méthode « `getIndividusList` » de la classe « `IndividuService` » après l'avoir instancié en utilisant le fichier généré. (Exemple : « `IndividuService service = new IndividuService("Mirko_DRAGIC.csv") ;` » si votre fichier s'appelle « `Mirko_DRAGIC.csv` »)
3. Le test doit vérifier le nombre d'individus de la liste
4. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
5. Cette méthode devra être commentée (pas besoin de faire un roman)
6. Commiter et vérifier l'exécution de la chaine CI/CD

Test des noms des 5 premiers individus chargés depuis le fichier CSV

1. Toujours dans cette classe de test, créer une méthode de test nommée « `testLoadedIndividusName` » sur la méthode « `getIndividusList` » de « `IndividuService` »

2. Le test doit vérifier le nom des 5 premiers individus de la liste
3. Le code de l'instanciation de la classe « `IndividuService` » ne doit pas être présent dans chaque méthode de test. Il faut qu'il soit une seule fois dans la classe de test pour être exécuté avant chaque test.
4. Cette méthode devra découper les parties « ARRANGE », « ACT » et « ASSERT ».
5. Cette méthode devra être commentée (pas besoin de faire un roman)
6. Commiter et vérifier l'exécution de la chaine CI/CD