# Tester c'est douter ?

#### Tests unitaires

- Consulter le fichier src/test/java/iut/rpg/ArmeTest.java
- Appliquer les consignes
- Lancer la suite de test via Maven : mvn test
- Lancer la suite de test via l'IDE
- Via l'IDE, poser un point d'arrêt et lancer le test en mode debug

• Consulter le fichier :

src/test/java/iut/rpg/JoueurTest.java

- Appliquer les consignes
- Utiliser @BeforeEach et @AfterEach pour simplifier les tests
- Utiliser AssertJ pour simplifier les tests <sup>[1]</sup>

• Consulter le fichier :

src/test/java/iut/rpg/TweetJoueurTest.java

• Appliquer les consignes

- Exécutez les tests avec Maven et visualisez le rapport HTML dans un navigateur
- Exécuter les tests avec l'IDE et visualisez le rapport

#### Tests d'intégration

#### Exercice 1 : Postman

- Lancer l'API "Rover" avec maven : mvn quarkus:dev
- Aller sur http://localhost:3000/q/swagger-ui/ pour voir l'interface Swagger
- Récupérer le fichier json sur http://localhost:3000/q/openapi
- Importer le fichier dans Postman

#### Exercice 2 : Postman

- Lancer des requêtes via Postman
- Ecrire les tests suivants <sup>[1]</sup>:
  - GET /rover retourne une liste non vide
  - GET /rover contient un rover du nom de "R2D2"

1. https://learning.postman.com/docs/tests-and-scripts/write-scripts/test-examples

## Exercice 3 : Playwright

- Lancer Keyboard Factory
  - npm install
  - ∎ npm run dev
- Lancer Playwright: npx playwright test

## Exercice 4 : Playwright

- Ecrire les tests suivants <sup>[1]</sup>:
  - La page d'accueil affiche un panier
  - Le panier est vide
  - On peut ajouter un article dans le panier

1. https://playwright.dev/docs/writing-tests