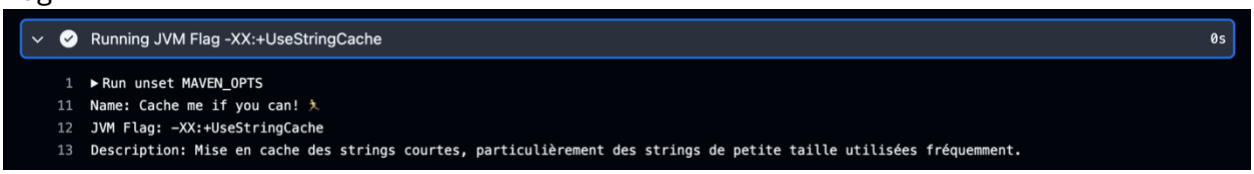
Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Les Flags JVM Choisis

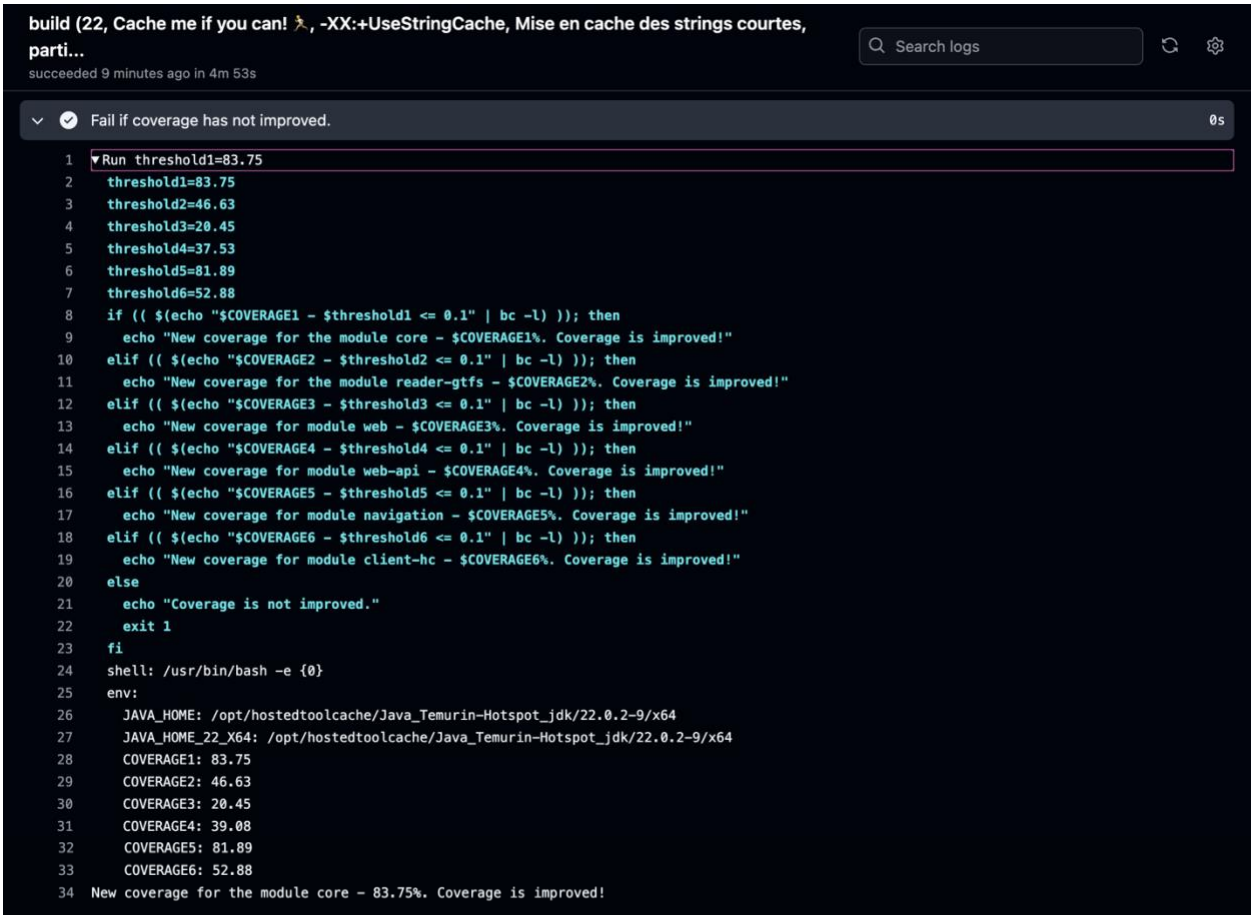
Flag n°1 -XX:+UseStringCache

Catégorie : Gestion des strings

Log :



Couverture :



Justification : Dans GraphHopper, qui manipule un grand nombre de chaînes de caractères identiques (noms de rues répétés,...) , ce flag permet d'optimiser l'utilisation de la mémoire. Au lieu de créer de nouvelles instances pour chaque chaîne identique, la JVM(Java Virtual Machine) réutilise les instances déjà existantes grâce au cache. Cette

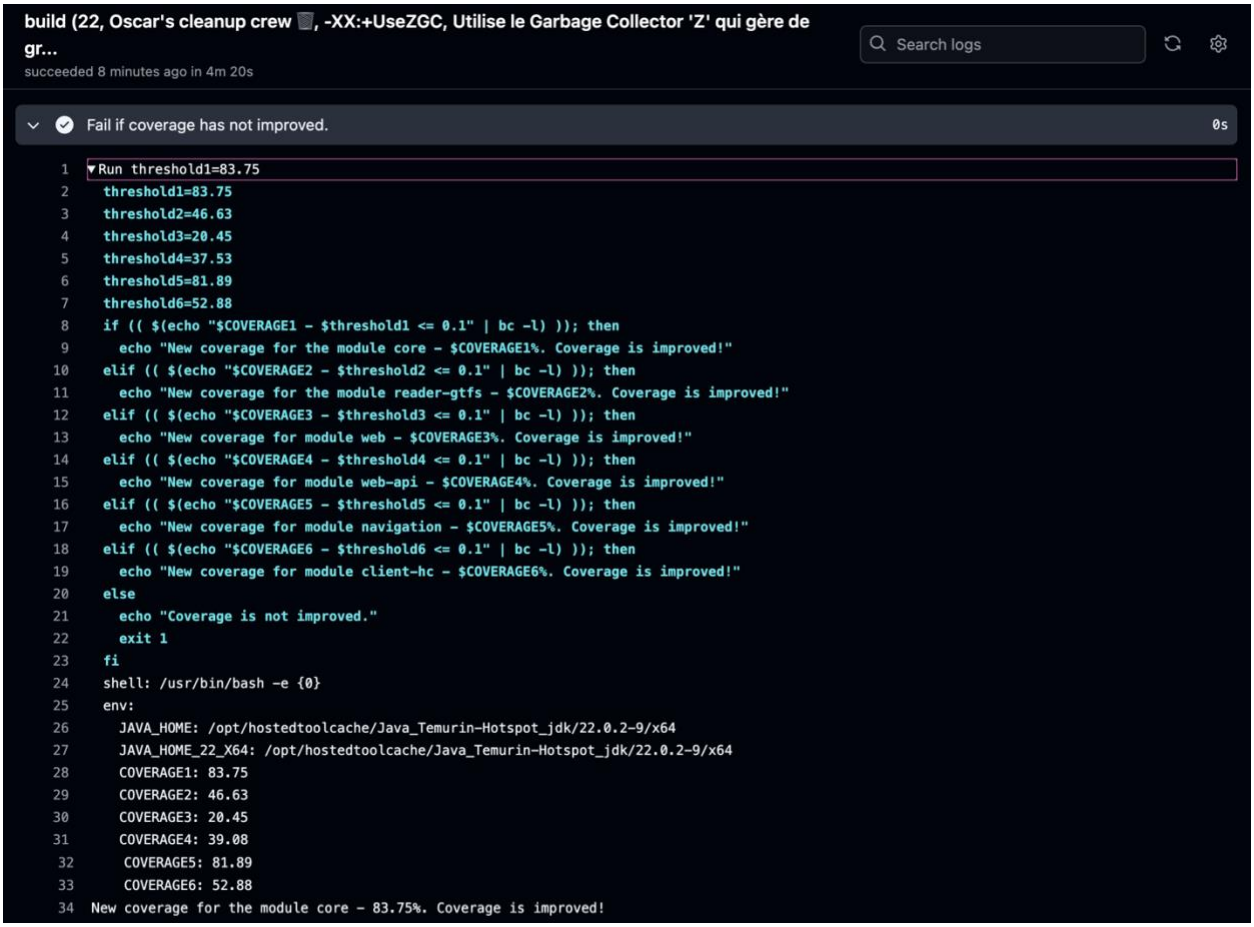
optimisation réduit non seulement l'empreinte mémoire mais améliore aussi les performances en évitant la création d'objets String redondants.

Flag n°2 -XX:+UseZGC

Catégorie : Garbage collector

Log :

Couverture :



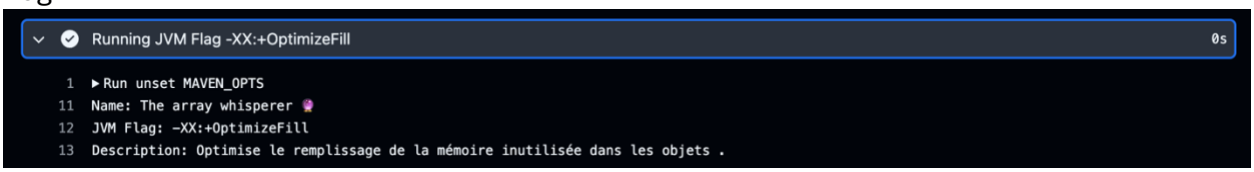
Justification : Comme Graph Hopper gère de nombreuses requêtes de routage et de grandes cartes (ex. des pays entiers), l'utilisation du ZGC minimise la latence et permet une expérience utilisateur fluide en éliminant les longues pauses de collecte de mémoire.

Hervé Ng'isse (20204609) et Brittany Curry-Sharples (20205096)

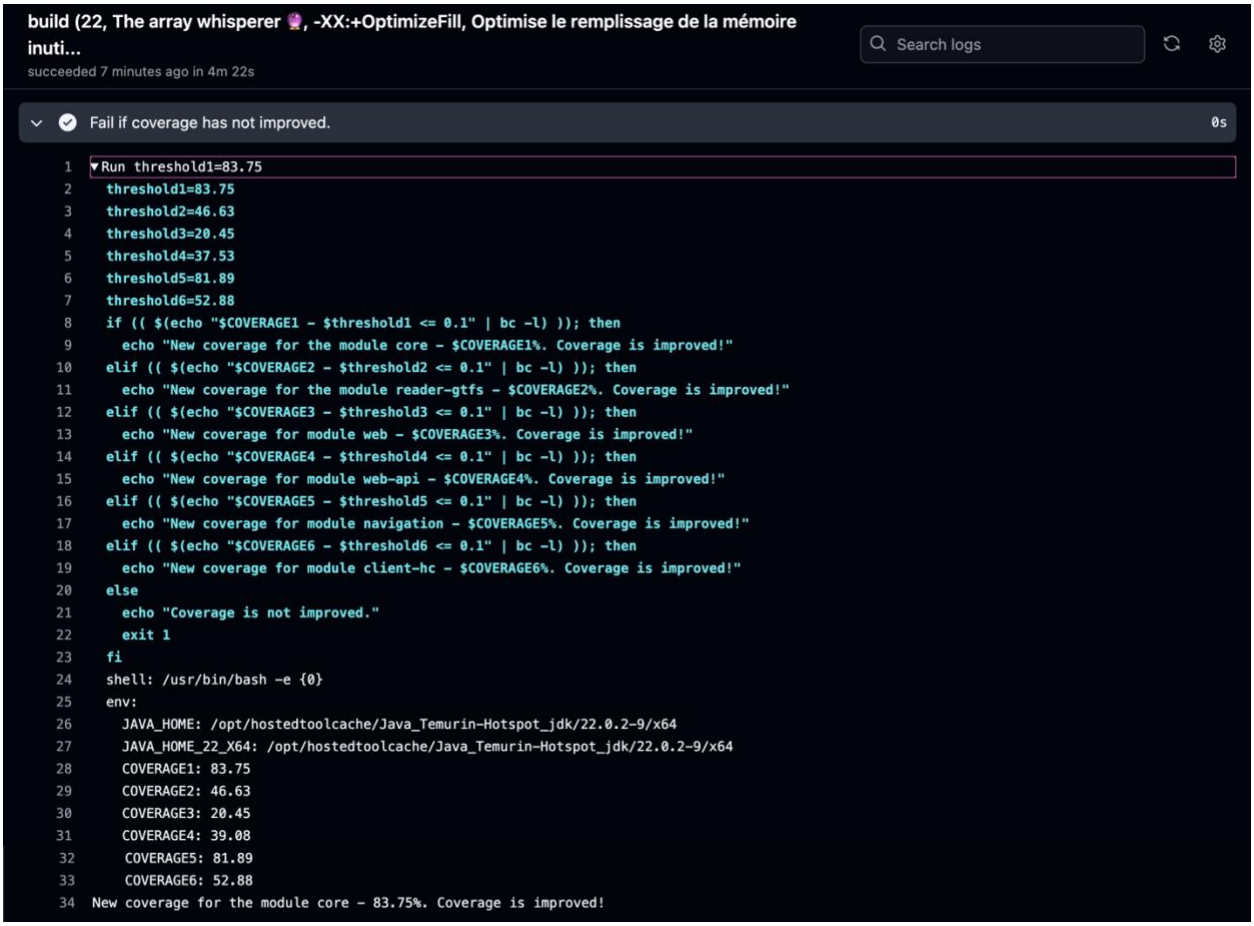
Flag n°3 : -XX:+OptimizeFill

Catégorie : Optimisation du mémoire et performance

Log :



Couverture :



Justification : Graph Hopper utilise des arrays extensivement pour stocker des graphes qui représentent une région donnée et pour exécuter des algorithmes sur ces graphes (ex. des algorithmes de routage). En optimisant les opérations de « fill » dans les arrays, la performance globale peut être améliorée.

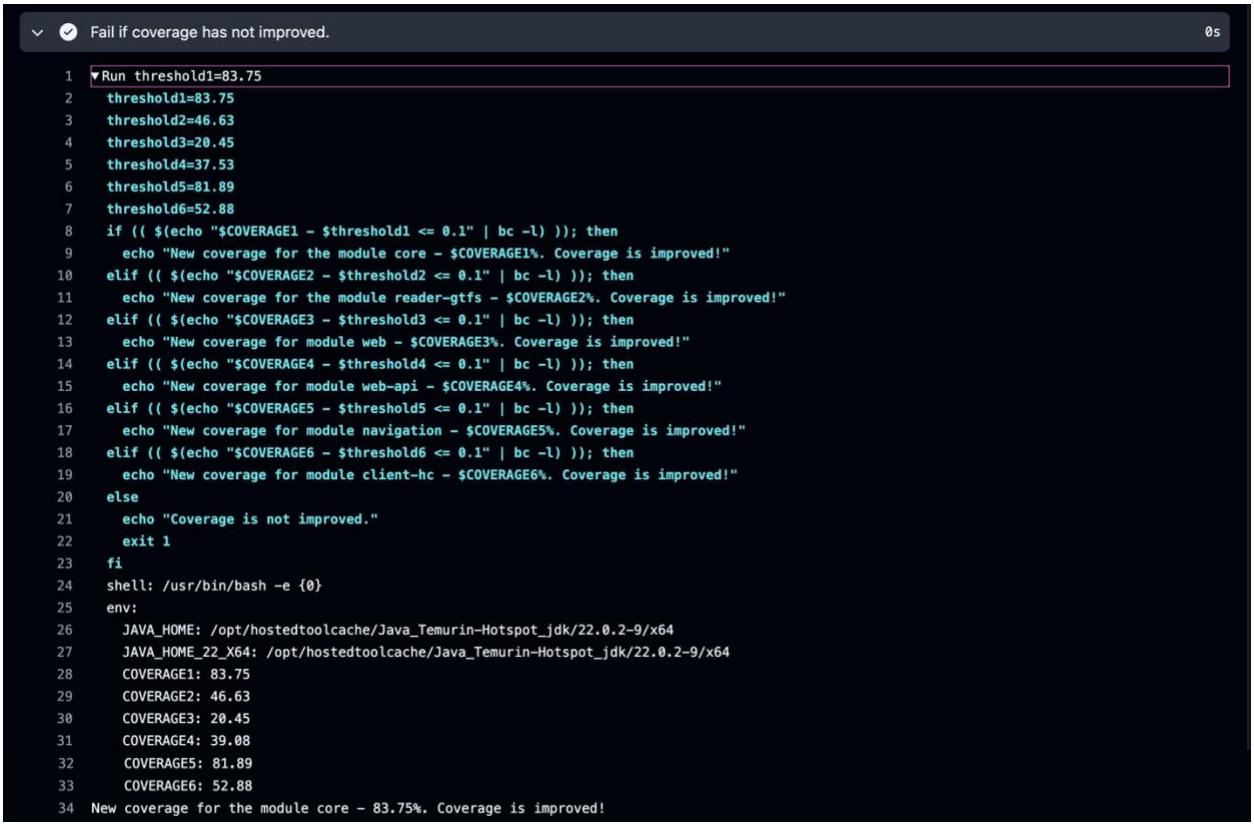
Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Flag n°4 -Dfile.encoding=UTF-8

Catégorie : Encodage

Log :

Couverture :



Justification : Dans GraphHopper, qui manipule des données cartographiques internationales, il y a forcément un risque de corruption des caractères spéciaux (comme les accents, caractères chinois) lors de la lecture des fichiers. Ce flag permet de garantir que tous les noms de rues, de villes et de points d'intérêt seront correctement lus et affichés.

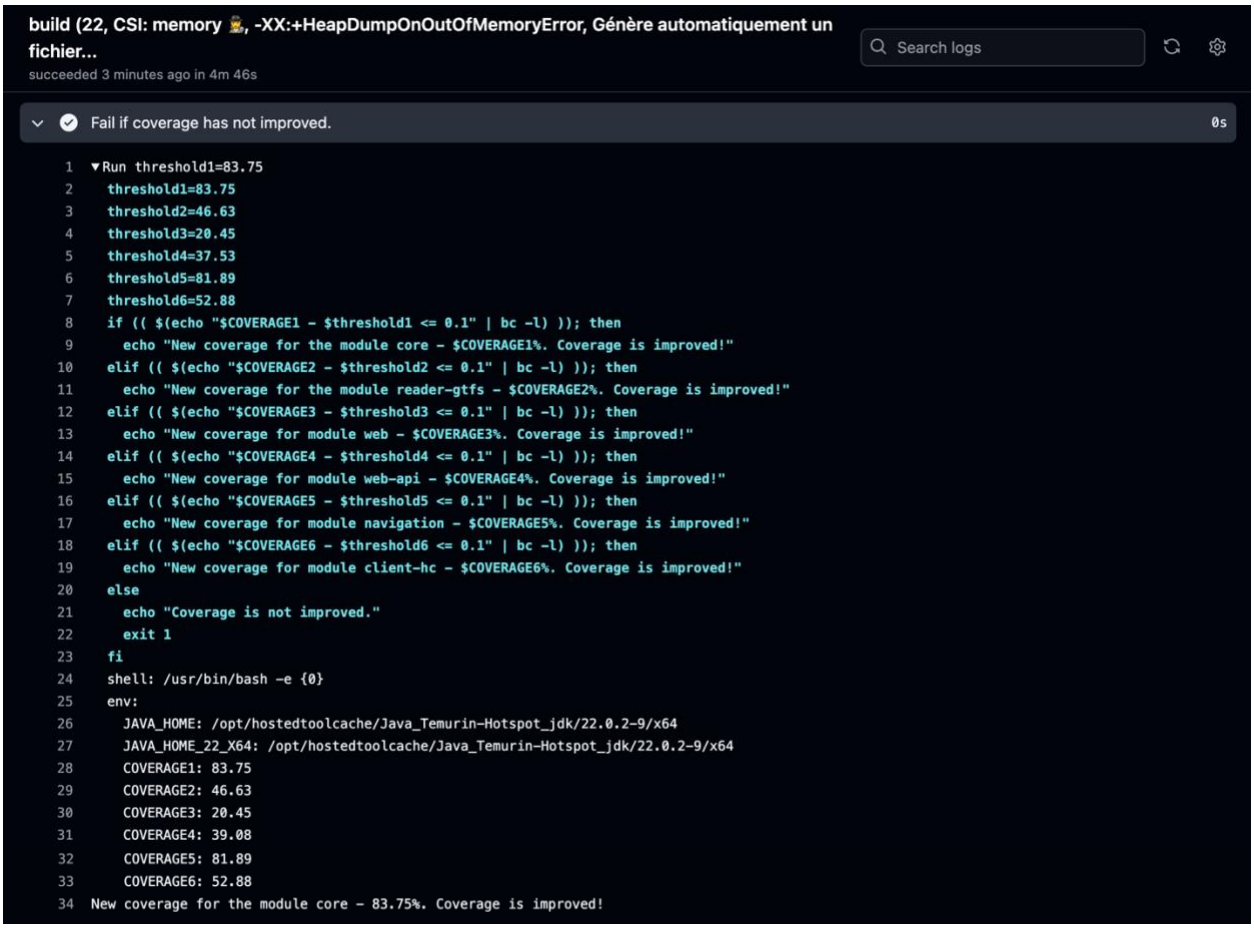
Flag n°5 -XX:+HeapDumpOnOutOfMemoryError

Catégorie : Débugage

Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Log :

Couverture :



Justification : GraphHopper manipule d'importants volumes de données cartographiques et effectue des calculs d'itinéraires complexes, ce qui en fait un candidat naturel aux erreurs OutOfMemory.ce flag est crucial car il génère automatiquement un dump de la heap lors de ces erreurs, permettant une analyse détaillée des problèmes de mémoire et facilitant leur résolution.

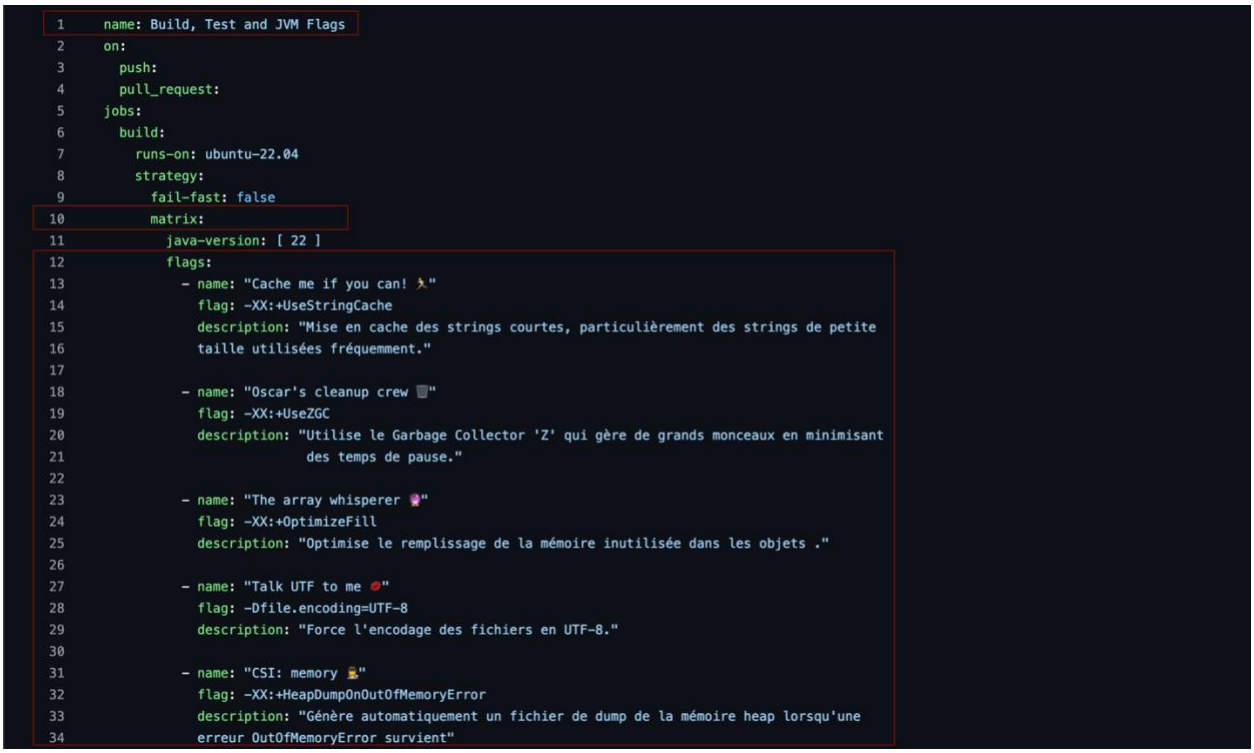
Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Changements apportés à la GitHub action

Tous les changements apportés à la GitHub action peuvent se trouver dans le document `build-test-jvmFlags.yml`. Les changements sont encerclés dans les captures d’écran ci-bas.

Le chemin vers ce fichier est comme suit : .github/workflows/build-test-jvmFlags.yml.

Le lien vers ce fichier sur notre repo GitHub peut se trouver ici.



```yaml
name: Build, Test and JVM Flags
on:
  push:
  pull_request:
jobs:
  build:
    runs-on: ubuntu-22.04
    strategy:
      fail-fast: false
      matrix:
        java-version: [ 22 ]
        flags:
          - name: "Cache me if you can! 🏃"
            flag: -XX:+UseStringCache
            description: "Mise en cache des strings courtes, particulièrement des strings de petite
            taille utilisées fréquemment."

          - name: "Oscar's cleanup crew 🗑️"
            flag: -XX:+UseZGC
            description: "Utilise le Garbage Collector 'Z' qui gère de grands monceaux en minimisant
                          des temps de pause."

          - name: "The array whisperer 🔮"
            flag: -XX:+OptimizeFill
            description: "Optimise le remplissage de la mémoire inutilisée dans les objets ."

          - name: "Talk UTF to me 💋"
            flag: -Dfile.encoding=UTF-8
            description: "Force l'encodage des fichiers en UTF-8."

          - name: "CSI: memory 🕵️"
            flag: -XX:+HeapDumpOnOutOfMemoryError
            description: "Génère automatiquement un fichier de dump de la mémoire heap lorsqu'une
            erreur OutOfMemoryError survient"
```

```yaml
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-java@v3
        with:
          java-version: ${{ matrix.java-version }}
          distribution: temurin
      - name: Cache Maven artifacts
        uses: actions/cache@v3
        with:
          path: ~/.m2/repository
          key: ${{ runner.os }}-maven-${{ hashFiles('**/pom.xml') }}
          restore-keys: |
            ${{ runner.os }}-maven-
      - name: Cache node
        uses: actions/cache@v3
        with:
          path: web-bundle/node
          key: ${{ runner.os }}-node-${{ hashFiles('**/pom.xml') }}
          restore-keys: |
            ${{ runner.os}}-node-
      - name: Cache node_modules
        uses: actions/cache@v3
        with:
          path: web-bundle/node_modules
          key: ${{ runner.os }}-node-${{ hashFiles('**/pom.xml', '**/package.json') }}
          restore-keys: |
            ${{ runner.os}}-node_modules-
      - name: Running JVM Flag ${{matrix.flags.flag}}
        run: |
          unset MAVEN_OPTS
          echo "Name: ${{matrix.flags.name}}"
          echo "JVM Flag: ${{matrix.flags.flag}}"
          echo "Description: ${{matrix.flags.description}}"
          export MAVEN_OPTS="${{matrix.flags.flag}}"

      - name: Build ${{ matrix.java-version }}
        run: mvn -B clean verify

      - name: Get JaCoCo Coverage
        id: COVERAGE
        run: |
          coverage1=$(python3 config/coverage.py core/target/site/jacoco/jacoco.csv)
          echo "COVERAGE1=$coverage1" >> $GITHUB_ENV
          coverage2=$(python3 config/coverage.py reader-gtfs/target/site/jacoco/jacoco.csv)
          echo "COVERAGE2=$coverage2" >> $GITHUB_ENV
          coverage3=$(python3 config/coverage.py web/target/site/jacoco/jacoco.csv)
          echo "COVERAGE3=$coverage3" >> $GITHUB_ENV
          coverage4=$(python3 config/coverage.py web-api/target/site/jacoco/jacoco.csv)
          echo "COVERAGE4=$coverage4" >> $GITHUB_ENV
          coverage5=$(python3 config/coverage.py navigation/target/site/jacoco/jacoco.csv)
          echo "COVERAGE5=$coverage5" >> $GITHUB_ENV
          coverage6=$(python3 config/coverage.py client-hc/target/site/jacoco/jacoco.csv)
          echo "COVERAGE6=$coverage6" >> $GITHUB_ENV
      - name: Fail if coverage has not improved.
        run: |
          threshold1=83.75
          threshold2=46.63
          threshold3=20.45
          threshold4=37.53
          threshold5=81.89
          threshold6=52.88
          if (( $(echo "$COVERAGE1 - $threshold1 <= 0.1" | bc -l) )); then
            echo "New coverage for the module core - $COVERAGE1%. Coverage is improved!"
          elif (( $(echo "$COVERAGE2 - $threshold2 <= 0.1" | bc -l) )); then
            echo "New coverage for the module reader-gtfs - $COVERAGE2%. Coverage is improved!"
          elif (( $(echo "$COVERAGE3 - $threshold3 <= 0.1" | bc -l) )); then
            echo "New coverage for module web - $COVERAGE3%. Coverage is improved!"
          elif (( $(echo "$COVERAGE4 - $threshold4 <= 0.1" | bc -l) )); then
            echo "New coverage for module web-api - $COVERAGE4%. Coverage is improved!"
          elif (( $(echo "$COVERAGE5 - $threshold5 <= 0.1" | bc -l) )); then
            echo "New coverage for module navigation - $COVERAGE5%. Coverage is improved!"
          elif (( $(echo "$COVERAGE6 - $threshold6 <= 0.1" | bc -l) )); then
            echo "New coverage for module client-hc - $COVERAGE6%. Coverage is improved!"
          else
            echo "Coverage is not improved."
            exit 1
          fi
```

Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

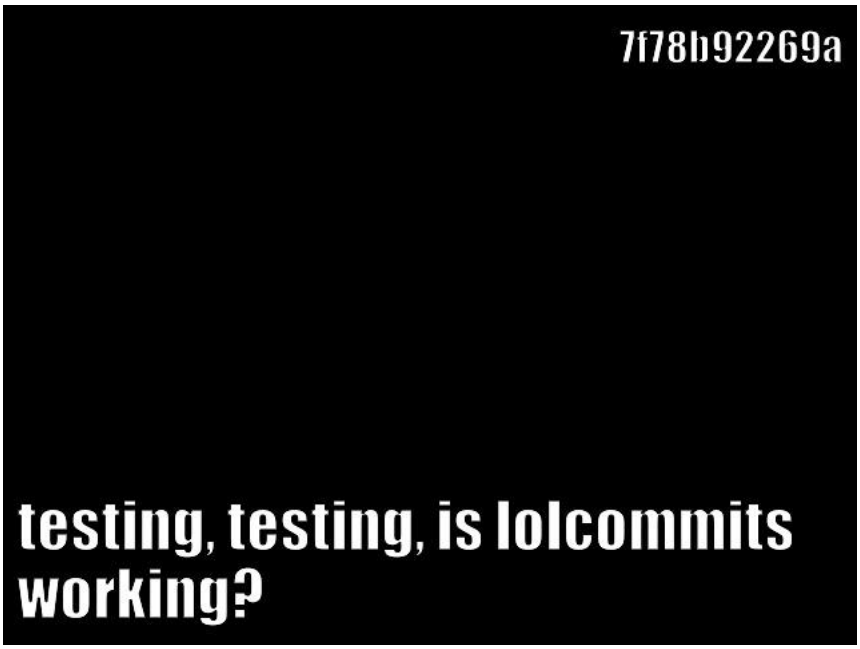
L’élément d’humour

L’élément d’humour qu’on a inclut est les noms qu’on a donnés au JVM flags, svp voir la capture d’écran ci-bas.

```
12          flags:
13            - name: "Cache me if you can! 🏃"
14              flag: -XX:+UseStringCache
15              description: "Mise en cache des strings courtes, particulièrement des strings de petite
16              taille utilisées fréquemment."
17
18            - name: "Oscar's cleanup crew 🗑️"
19              flag: -XX:+UseZGC
20              description: "Utilise le Garbage Collector 'Z' qui gère de grands monceaux en minimisant
21                            des temps de pause."
22
23            - name: "The array whisperer 🔮"
24              flag: -XX:+OptimizeFill
25              description: "Optimise le remplissage de la mémoire inutilisée dans les objets ."
26
27            - name: "Talk UTF to me 💋"
28              flag: -Dfile.encoding=UTF-8
29              description: "Force l'encodage des fichiers en UTF-8."
30
31            - name: "CSI: memory 🔍"
32              flag: -XX:+HeapDumpOnOutOfMemoryError
33              description: "Génère automatiquement un fichier de dump de la mémoire heap lorsqu'une
34              erreur OutOfMemoryError survient"
```

Bonus : lolcommits

Quelques de nos lolcommits :



Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)

Hervé Ng’isse (20204609) et Brittany Curry-Sharples (20205096)