

LOG3430 - MÉTHODES DE TEST ET DE VALIDATION DU LOGICIEL

LABORATOIRE 1

COUVERTURE DE TEST

Département de génie informatique et de génie logiciel
École Polytechnique de Montréal



Automne 2024

1 Introduction

Dans ce laboratoire, vous devrez apprendre à utiliser des outils de couverture de test et à analyser leurs résultats.

La couverture de test est une mesure de la quantité de code exécutée lors de l'exécution d'une suite de tests particulière. Les outils de couverture de test sont couramment utilisés dans le cadre du processus d'intégration continue pour garantir la qualité du code. Vous trouverez ci-dessous une partie des résultats de couverture générés par un outil de couverture (c'est-à-dire `pytest-cov` ou `coverage.py`).

Module	statements	missing	excluded	branches	partial	coverage
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_pytest_runner.py	33	3	0	19	7	81%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_runner_tools.py	173	48	0	54	10	71%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_serial_tree_manager.py	56	7	0	26	7	80%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_utils.py	71	34	0	28	0	49%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/teamcity/_init_.py	6	0	0	0	0	100%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/teamcity/common.py	98	60	0	45	3	34%
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/teamcity/diff_tools.py	62	36	0	19	2	37%

À partir du rapport, vous pouvez savoir combien de lignes/branches/instructions sont couvertes. Ici, nous utiliserons l'outil pour analyser des projets Python.

2 Objectifs

Les objectifs généraux de ce laboratoire sont :

1. Apprendre à utiliser un IDE (PyCharm) pour déterminer la couverture de code d'une suite de tests.
2. Apprendre à utiliser des arguments de ligne de commande pour tirer parti d'une bibliothèque telle que `coverage.py` pour générer un rapport de couverture de code.
3. Pratiquer la conception de jeux de tests satisfaisants divers critères de couvertures.

3 Critères de couverture de base

Il existe plusieurs framework des tests unitaires spécifiques à chaque langage. Parmi les frameworks les plus populaires on trouve JUnit pour Java, **Unittest** et **PyTest** pour **Python**, RSpec pour Ruby , Karma, Jasmine, Mocha, Chai et Sinon pour JavaScript, etc.

Pour mesurer le pourcentage de code exécuté par une suite de tests unitaires, un ou plusieurs critères de couverture sont utilisés. Ici, nous introduisons 4 critères de couverture de base qui ont été vus en classe.

- **Couverture fonction/méthode**
- **Couverture d'instructions (statement)**
- **Couverture de branches**
- **Couverture de conditions**

Vous pouvez consulter ces liens pour plus de détails :

- <https://docs.python.org/3/library/unittest.html>
- <https://coverage.readthedocs.io/en/7.4.0/>
- <https://pytest-cov.readthedocs.io/en/latest/>

Les outils

Il existe plusieurs outils pour aider à analyser la couverture de code Python.

- **pytest-cov** - plugin pour le framework de test Python appelé Pytest
- **coverage.py** - module Python qui permet de mesurer la couverture de code
- **PyCharm** - IDE qui inclut un outil de visualisation de couverture de code
- Visual Studio Code - autre IDE qui à en option un outil de visualisation de couverture de code (Coverage Gutters)

Outils Requis

Veuillez vous assurer que **Python** et **PyCharm Professional** (vous avez accès à une licence gratuite en tant qu'étudiants : <https://www.jetbrains.com/pycharm/>) sont installés sur votre ordinateur.

Système d'exploitation : Linux/macOS/Windows

Python : version 3.9+

Les tâches

1. Obtenez une copie locale du code source de la bibliothèque "Flask" (<https://github.com/pallets/flask>) et installez les dépendances nécessaires avec les commandes suivante :
 - pip install -e .
 - pip install -r requirements/tests.txt
 - pip install pytest-cov
2. Répondez aux questions suivantes :

Important ! : Il est nécessaire d'inclure dans le rapport les captures d'écran illustrant les résultats des rapports de couverture pour chaque question. Pour la question 7, veuillez également intégrer des captures d'écran du code que vous avez rédigé.

- (a) Question 1 : Utilisez l'IDE PyCharm pour générer un rapport de couverture de code (Run 'pytest in tests' with Coverage, puis Generate Coverage Report). Quelle est la couverture totale du code pour Flask ? Vous devez aussi soumettre le dossier complet du rapport de couverture sous la forme de fichier zip nommé Q1.zip.
- (b) Question 2 : Utilisez l'IDE PyCharm pour activer la couverture des branches (Settings - Build, Execution, Deployment - Coverage) et générer un nouveau rapport de couverture de code. Quelle est la couverture totale du code pour Flask ? Y a-t-il une différence avec le rapport précédent ? Expliquez pourquoi il y a/n'y a pas de différence. Vous devez aussi soumettre le dossier complet du rapport de couverture sous la forme de fichier zip nommé Q2.zip.
- (c) Question 3 : En utilisant la méthode de votre choix, excluez un fichier du processus de génération du rapport de test (sans le supprimer). Quel fichier avez-vous exclu (nom et répertoire du fichier) et comment l'avez-vous exclu du processus ? Comment les résultats changent-ils par rapport au dernier rapport (conservez l'option

- pour avoir la couverture des branches) ? Vous devez aussi soumettre le dossier complet du rapport de couverture sous la forme de fichier zip nommé Q3.zip.
- (d) Question 4 : Veuillez expliquer, étape par étape, comment utiliser coverage.py pour calculer la couverture de test de Flask sans IDE (utilisez l'option pour branch coverage). Justifiez chaque terme de la commande que vous avez utilisée. Vous devez aussi soumettre le dossier complet du rapport de couverture sous la forme de fichier zip nommé Q4.zip.
 - (e) Question 5 : Veuillez expliquer, étape par étape, comment utiliser pytest-cov pour calculer la couverture de test de Flask sans IDE (utilisez l'option pour branch coverage). Justifiez chaque terme de la commande que vous avez utilisée. Vous devez aussi soumettre le dossier complet du rapport de couverture sous la forme de fichier zip nommé Q5.zip.
 - (f) Question 6 : Veuillez analyser et discuter les éventuelles divergences en termes de couverture de lignes entre les deux méthodes récentes abordées. Quels sont les types de lignes couvertes par l'un mais pas par l'autre ? (Nommez-en 3). Expliquez pourquoi ces lignes sont couvertes/ignorées par ces outils ?
Indice : Comparez deux rapports de couverture que vous avez généré dans les questions précédentes pour un même fichier. (Veuillez soumettre des captures d'écran appuyant vos réponses).
 - (g) Question 7 : Vous devez rédiger deux nouveaux tests afin d'améliorer la couverture de test de la bibliothèque Flask. Veuillez détailler les tests que vous avez créés. Utilisez la méthode de votre choix, parmi celles mentionnées dans les questions précédentes relatives à la couverture de test, pour démontrer quel(s) critère(s) de couverture de test vos nouveaux tests ont amélioré(s). Vous devez aussi soumettre le dossier complet du rapport de couverture ainsi que le repo Flask complet sous la forme de fichier zip nommé Q7.zip.

4 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire. Le rapport doit contenir :
 - Vos réponses à la question 1 : (2 points).
 - Vos réponses à la question 2 : (2 points).
 - Vos réponses à la question 3 : (3 points).
 - Vos réponses à la question 4 : (3 points).
 - Vos réponses à la question 5 : (3 points).
 - Vos réponses à la question 6 : (3 points).
 - Vos réponses à la question 7 : (3 points).
 - Qualité du rapport : (1 point).
- Le dossier COMPLET contenant les rapports de couverture (comme indiqué dans chaque question), ainsi que le code source nécessaire pour effectuer les tests pour la question 7. N'oubliez pas de bien commenter votre code.
- **L'absence d'un rapport de couverture de code ou d'une capture d'écran pour une question qui en fait la demande entraînera une non-notation de**

cette question.

Le tout à remettre dans une seule archive **zip** avec le titre matricule1_matricule2_lab1.zip sur Moodle. Seulement une personne de l'équipe doit remettre le travail.

Le rapport doit contenir le titre et numéro du laboratoire, les noms et matricules des coéquipiers ainsi que le numéro du groupe.

5 Information importante

1. Consultez le site Moodle du cours pour la date et l'heure limites de remise des fichiers.
Consultez l'onglet choix d'équipe afin d'avoir accès à la remise, le travail doit se faire en équipe de 2 étudiants.
2. Un retard de $[0, 24h]$ sera pénalisé de 10%, de $[24h, 48h]$ de 20% et de plus de 48h de 50%.
3. Aucun plagiat n'est toléré. Vous devez soumettre uniquement le code et les rapports de couverture de code réalisé par les membres de votre équipe.