

# 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



**POLYTECHNIQUE  
MONTREAL**

Automne 2022

# 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/fr/3.8/library/unittest.html>
- <https://coverage.readthedocs.io/en/coverage-5.3/index.html>

— <https://pytest-cov.readthedocs.io/en/latest/>

## Les outils

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

- **pytest-cov** - librairie de couverture de code
- **coverage.py** - autre librairie de 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** (vous avez accès à une licence gratuite en tant qu'étudiants) sont installés sur votre ordinateur.

**Système d'exploitation** : Linux/macOS/Windows

**Python** : version 3.7 (ou+)

## Les tâches

1. Obtenez une copie locale du code source de la bibliothèque "Requests" (v2.28.1).  
<https://github.com/psf/requests>
2. Répondre aux questions suivantes :  
**Important !** Dans votre rapport du TP pour chaque question il faut ajouter les captures d'écran, qui montrent les résultats obtenus dans le rapport de couverture. Pour les question 6 et 7 il faut ajouter les captures d'écran avec le code que vous avez ajouté.
  - (a) Question 1 : En générant un rapport de couverture de code à l'aide de l'IDE PyCharm, quelle est la couverture totale du code ( statement coverage, missing, excluded, coverage %) pour Requests v2.28.1 ? Vous devez aussi soumettre le rapport de couverture sous forme de fichier zip nommé Q1.zip.
  - (b) Question 2 : Utilisez l'IDE PyCharm pour activer la couverture des branches. En générant un nouveau rapport de couverture de code à l'aide de l'IDE PyCharm, quelle est la couverture totale du code (files, lines, statements, missing, excluded, branches, partial coverage %) pour Requests v2.28.1 ? 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 rapport de couverture sous forme de fichier zip nommé Q2.zip.
  - (c) Question 3 : Vous devez exclure un fichier de code source du processus de génération de rapport de test. Quel fichier avez-vous exclu (nom et répertoire du fichier) ? Comment l'avez-vous exclu du processus ? Comment les résultats changent-ils par rapport au dernier rapport (conserver l'option pour avoir la couverture des

branches)? Vous devez soumettre le rapport généré sous 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 Requests sans IDE (utiliser l'option pour branch coverage). Vous devez soumettre le rapport généré par coverage.py sous forme de fichier zip nommé Q4.zip.
- (e) Question 5 : Veuillez expliquer, étape par étape, comment utiliser pytest pour calculer la couverture de test de Requests sans IDE (utiliser l'option pour branch coverage). Vous devez soumettre le rapport généré sous forme de fichier zip nommé Q5.zip.
- (f) Question 6 : Vous devez créer deux nouveaux tests pour augmenter la couverture de test de Requests. Quels tests avez-vous écrit ? En utilisant l'une des deux méthodes de couverture de test des questions précédentes (PyCharm ou Coverage.py ou pytest), montrez quels critère(s) de couverture de test vos tests ont augmentés. Vous devez soumettre un zip «Q6.zip» du repo Requests complet dans le quel vous avez ajouté vos tests.
- (g) Question 7 : Veuillez discuter s'il y a des lignes qui sont couvertes par un outil mais pas par l'autre et votre processus d'enquête sur la raison. Le processus de vérifications devrait être automatisé. Comparer 2 outils serait suffisant. On recommande coverage.py et pytest -cov. Il faut considérer la couverture avec branches. Il faut que à la sortie votre script génère le json avec la structure suivante :

```
{  
    "requests\\__init__.py": [0, 1, 2, 3, 4],  
    "requests\\adapters.py": [0, 1, 3, 6, 10],  
    "requests\\api.py": [4, 6, 9]  
}
```

(Veuillez soumettre tout le code source pour votre analyse automatisée dans un fichier zip «Q7.zip»).

Indice : il est possible que les tests ne s'exécutent pas à cause d'un problème de configuration. Si vous avez une erreur de BaseResponse, essayez de spécifier manuellement werkzeug==2.0.3 dans les requirements.txt.

## 4 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire. Le rapport doit contenir :
  - Vos réponses à la question 1 : (3 points).
  - Vos réponses à la question 2 : (3 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 : (4 points).
- Vos réponses à la question 7 : (4 points).
- Le dossier COMPLET contenant les rapports (comme indiqué dans chaque question), ainsi que le code source nécessaire pour effectuer les tests pour la question 6 et l'évaluation automatique pour la question 7. **N'oubliez pas de bien commenter votre code.** Si un rapport de couverture de code est manquant pour une question qui le demande, la question ne sera pas notée.

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.
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.