

1.1 Question 1

Exécutez Mutmut sur le projet ipinfo que vous avez téléchargé. Notez que cela nécessitera probablement de consulter la documentation de Mutmut. Combien de mutants ont été tués? Combien ont expiré (timed out) ? Combien étaient suspects, combien ont survécu ?

Commande : mutmut run

Le nombre de mutants tués est 116.

expirés est 0.

suspects est 0.

survivants est 236.

```
Legend for output:
❑ Killed mutants.    The goal is for everything to end up in this bucket.
🕒 Timeout.          Test suite took 10 times as long as the baseline so were killed.
❑ Suspicious.       Tests took a long time, but not long enough to be fatal.
❑ Survived.         This means your tests need to be expanded.
❑ Skipped.          Skipped.

mutmut cache is out of date, clearing it...
1. Running tests without mutations
.: Running...Done

2. Checking mutants
🔍 352/352 ❑ 116 🕒 0 ❑ 0 ❑ 236 ❑ 0
```

1.2 Question 2

Générez un rapport html à l'aide de Mutmut (mutmut html). À l'aide d'un outil de couverture de code de votre choix, générez un rapport de couverture de code avec couverture d'instruction (statement coverage). En utilisant les deux rapports, quelle est la couverture d'instruction du fichier avec le plus de mutants survivant (et quel est le nom du fichier ?) ? Quelle est la couverture de d'instruction du fichier avec le moins de mutants survivants (quel est le nom du fichier ?) ? Pour obtenir tous les points pour cette question, vous devez soumettre les deux rapports dans un fichier Q2.zip en plus des réponses à cette question.

La couverture d'instruction du fichier `ipinfo\handler_async.py` a le plus de mutants survivants (87 mutants survivants) et une couverture de 31%.

La couverture d'instruction du fichier `ipinfo\cache\default.py` a le moins de mutants survivants (0 mutant survivant) et une couverture de 92%.

Mutation testing report

Killed 116 out of 352 mutants

File	Total	Killed	% killed	Survived
ipinfo\bogon.py	58	56	96.55	2
ipinfo\details.py	4	3	75.00	1
ipinfo\handler.py	110	26	23.64	84
ipinfo\handler_async.py	98	11	11.22	87
ipinfo\handler_utils.py	75	19	25.33	56
ipinfo\version.py	2	0	0.00	2
ipinfo\cache\default.py	1	1	100.00	0
ipinfo\cache\interface.py	4	0	0.00	4

Coverage report: 58%

coverage.py v7.1.0, created at 2023-02-13 10:38 -0500

Module	statements	missing	excluded	coverage
ipinfo__init__.py	6	0	0	100%
ipinfo\bogon.py	7	1	0	86%
ipinfo\cache__init__.py	0	0	0	100%
ipinfo\cache\default.py	13	1	0	92%
ipinfo\cache\interface.py	14	4	0	71%
ipinfo\details.py	10	0	0	100%
ipinfo\exceptions.py	4	0	0	100%
ipinfo\handler.py	115	65	0	43%
ipinfo\handler_async.py	125	86	0	31%
ipinfo\handler_utils.py	46	15	0	67%
ipinfo\version.py	1	0	0	100%
tests__init__.py	0	0	0	100%
tests\default_cache_test.py	13	0	0	100%
tests\details_test.py	19	0	0	100%
tests\handler_async_test.py	63	24	0	62%
tests\handler_test.py	62	18	0	71%
tests\init_test.py	9	0	0	100%
Total	507	214	0	58%

1.3 Question 3

Trouvez quatre mutants créés à partir de quatre types de mutateurs différents, affichez ces mutants et nommez le type de chaque mutant.

Le mutant 3 est de type Return Values Mutator (RETURN_VALS) :

Mutant 3

```
--- ipinfo\bogon.py
+++ ipinfo\bogon.py
@@ -5,7 +5,7 @@
     for network in BOGON_NETWORKS:
         if IP(ip_address) in network:
             return True
-    return False
+    return True
```

```
BOGON_NETWORKS = [
```

Le mutant 152 est de type Conditionals Boundary Mutator (CONDITIONALS_BOUNDARY):

Mutant 152

```
--- ipinfo\handler.py
+++ ipinfo\handler.py
@@ -232,7 +232,7 @@
     # quit if total timeout is reached.
     if (
         timeout_total is not None
-        and time.time() - start_time > timeout_total
+        and time.time() - start_time >= timeout_total
     ):
         return handler_utils.return_or_fail(
             raise_on_fail, TimeoutExceededError(), result
```

Le mutant 157 est de type Negate Conditionals Mutator (NEGATE_CONDITIONALS) :

Mutant 157

```
--- ipinfo\handler.py
+++ ipinfo\handler.py
@@ -250,7 +250,7 @@

        # fail on bad status codes
    try:
-        if response.status_code == 429:
+        if response.status_code != 429:
            raise RequestQuotaExceededError()
        response.raise_for_status()
    except Exception as e:
```

Le mutant 158 est de type Inline Constant Mutator (INLINE_CONST) :

Mutant 158

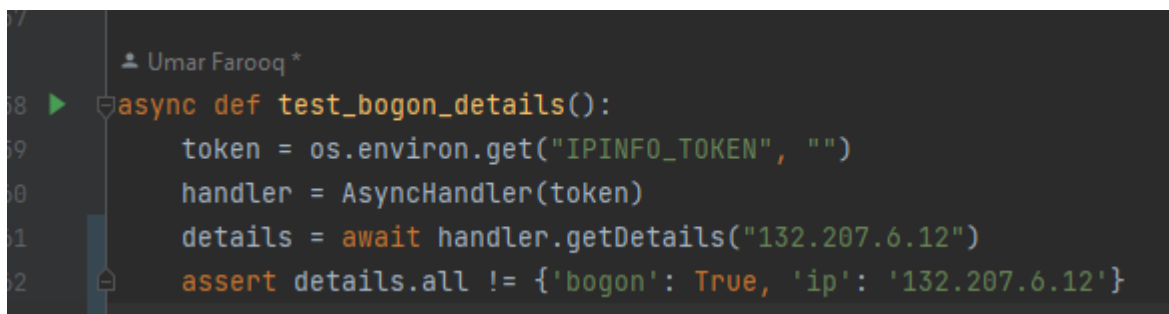
```
--- ipinfo\handler.py
+++ ipinfo\handler.py
@@ -250,7 +250,7 @@

        # fail on bad status codes
    try:
-        if response.status_code == 429:
+        if response.status_code == 430:
            raise RequestQuotaExceededError()
        response.raise_for_status()
    except Exception as e:
```

1.4 Question 4

Corrigez les tests d'ipinfo pour tuer les mutants que vous avez identifiés à la question 3. Quels tests avez-vous modifiés (donnez le chemin des fichiers et le nom des tests), et pourquoi avez-vous fait les modifications que vous avez faites ? Relancez Mutmut, quels sont les nouveaux résultats ? Pour cette question, vous devez fournir une copie complète du repo de ipinfo après vos modifications.

Pour le mutant 3, nous avons modifié le test `test_bogon_details()` dans le fichier `handler_async_test.py`. Nous avons modifié l'adresse IP en remplaçant `127.0.0.1` par `132.207.6.12`. Nous avons effectué cette modification afin de tester une adresse IP qui n'est pas réservée, contrairement à la précédente.



```
Umar Farooq *
38 ▶ async def test_bogon_details():
39     token = os.environ.get("IPINFO_TOKEN", "")
40     handler = AsyncHandler(token)
41     details = await handler.getDetails("132.207.6.12")
42     assert details.all != {'bogon': True, 'ip': '132.207.6.12'}
```

Pour le mutant 152, nous avons ajouté le test `test_get_batch_details_total_timeout_not_called(batch_size)` dans le fichier `handler_test.py`. Nous avons mock la valeur de retour de `time.time()` afin que la 1^{er} appelée soit 0 et la 2^e appelée soit 0.001. On mock le status response de la méthode POST afin de soulever l'erreur `RequestQuotaExceededError` pour arrêter la boucle for. Nous avons effectué ce test pour vérifier qu'on appelle pas `TimeoutExceededError` au cas notre `time.time() - start_time` est inférieur ou égal au `timeout_total`.

```
Uman Shahzad *
6 @pytest.mark.parametrize("batch_size", [1, 2])
7 def test_get_batch_details_total_timeout_not_called(batch_size):
8     # manually assigning variables because of _prepare_batch_test skipping the test
9     ips = ["1.1.1.1", "8.8.8.8", "9.9.9.9"]
10    token = os.environ.get("IPINFO_TOKEN", "")
11    handler = Handler(token)
12    with patch("time.time", side_effect=[0, 0.001]):
13        mock_response = MagicMock()
14        mock_response.status_code = 429
15        with patch('requests.post', return_value=mock_response):
16            # have to raise RequestQuotaExceededError
17            with pytest.raises(ipinfo.exceptions.RequestQuotaExceededError):
18                handler.getBatchDetails(
19                    ips, batch_size=batch_size, timeout_total=0.001
20                )
21
```

Pour les mutants 157 et 158, nous avons ajouté le test `test_get_batch_details_response(batch)` dans le fichier `handler_test.py`. Nous avons mock la valeur de retour afin de vérifier que l'on appelle bien `RequestQuotaExceededError`. Nous avons effectué cet ajout pour tester si `RequestQuotaExceededError` est bien appelé seulement par le status code 429.

```
new *
145 @pytest.mark.parametrize("batch_size", [1, 2])
146 def test_get_batch_details_response(batch_size):
147     # manually assigning variables because of _prepare_batch_test skipping the test
148     ips = ["1.1.1.1", "8.8.8.8", "9.9.9.9"]
149     token = os.environ.get("IPINFO_TOKEN", "")
150     handler = Handler(token)
151     mock_response = MagicMock()
152     mock_response.status_code = 429
153     with patch('requests.post', return_value=mock_response):
154         with pytest.raises(ipinfo.exceptions.RequestQuotaExceededError):
155             handler.getBatchDetails(
156                 ips, batch_size=batch_size
157             )
158
```

Le nombre de mutants tués est 131.
expirés est 0.
suspects est 0.
survivants est 221.

Legend for output:

💩 Killed mutants.	The goal is for everything to end up in this bucket.
🕒 Timeout.	Test suite took 10 times as long as the baseline so were killed.
😬 Suspicious.	Tests took a long time, but not long enough to be fatal.
😬 Survived.	This means your tests need to be expanded.
🚫 Skipped.	Skipped.

mutmut cache is out of date, clearing it...

1. Running tests without mutations

.: Running...Done

2. Checking mutants

.: 352/352 💩 131 🕒 0 😬 0 😬 221 🚫 0