**Задание «Тестирование асинхронных методов и метрики»**

Максимальное кол-во баллов:

10 баллов

Крайний срок сдачи:

14.11.2024

Напишите тесты к API используя метрики покрытия и асинхронное программирование

Выполнить задание согласно презентации

Приложение из Булгакова типа "document". [Покрытие кода тестами.pptx](https://storage.yandexcloud.net/s3lxpbulgakov/public/documents/14299/btUupyVqEBtnIPVB5SZEaEgHVksTIwBUQdOtH9OU.pptx)

[Покрытие кода тестами.pptx](https://storage.yandexcloud.net/s3lxpbulgakov/public/documents/14299/btUupyVqEBtnIPVB5SZEaEgHVksTIwBUQdOtH9OU.pptx):

1 слайд:

Покрытие кода тестами

2 слайд:

* Установить командную строку для работы с Ubuntu
* Вызвать тестовое покрытие
* Проанализировать результаты

3 слайд:

Теперь вы можете установить все необходимое для запуска WSL с помощью одной команды. Откройте Командную строку PowerShell или Windows в режиме администратора, щелкнув правой кнопкой мыши и выбрав команду "Запуск от имени администратора", введите команду wsl --install, а затем перезапустите компьютер.

PowerShell

wsl --install

Эта команда включит функции, необходимые для запуска WSL и установки дистрибутива Ubuntu для Linux. (Этот дистрибутив по умолчанию можно изменить.)

Если вы используете более раннюю сборку или не хотите использовать команду для установки и вам нужны пошаговые инструкции, см. статью Действия по установке WSL вручную для старых версий.

При первом запуске недавно установленного дистрибутива Linux откроется окно консоли. Вам будет предложено подождать, пока файлы будут распакованы и сохранены на компьютере. Все будущие запуски должны занимать меньше секунды.

Примечание

--

Приведенная выше команда работает, только если WSL не установлена вообще. Если вы выполнили wsl install и видите текст справки WSL, попробуйте выполнить wsl --list --online, чтобы просмотреть список доступных дистрибутивов. Затем выполните wsl --install -d <DistroName> для установки дистрибутива. Сведения об удалении WSL см. в разделе Удаление устаревшей версии WSL или Отмена регистрации или удаление дистрибутива Linux.

4 слайд

Terminal

neon673@WIN-7H51NCBV4ID:/mnt/c/Users/user/PycharmProjects/selenium\_otus$

5 слайд

Покрытие кода является показателем того, какой процент тестируемого кода тестируется набором тестов. Когда вы запускаете тесты для проекта «Tasks», некоторые функции «Tasks» выполняются с каждым тестом, но не со всеми.

Инструменты покрытия кода отлично подходят для того, чтобы сообщить вам, какие части системы полностью пропущены тестами.

Coverage.py является предпочтительным инструментом покрытия Python, который измеряет покрытие кода.

Вы будете использовать его для проверки кода проекта Tasks с помощью pytest.

Что бы использовать coverage.py нужно его установить. Не помешает установить плагин под названием pytest-cov, который позволит вам вызывать coverage.py от pytest с некоторыми дополнительными опциями pytest. Поскольку coverage является одной из зависимостей pytest-cov, достаточно установить pytest-cov и он притянет за собой coverage.py:

6 слайд

$ pip install pytest-cov

Collecting pytest-cov

Using cached pytest\_cov-2.5.1-py2.py3-none-any.whl

Collecting coverage>=3.7.1 (from pytest-cov)

Using cached coverage-4.4.1-cp36-cp36m-macosx\_10\_10\_x86

Installing collected packages: coverage, pytest-cov

Successfully installed coverage-4.4.1 pytest-cov-2.5.1

7 слайд

Теперь, когда установлена следующая версия задач, можно запустить отчет о базовом покрытии:

$ cd /path/to/code/ch7/tasks\_proj\_v2

$ pytest--cov=src

======= test session starts ======

plugins: mock-1.6.2, cov-2.5.1

collected 62 items

tests/func/test\_add.py

tests/func/test\_add\_variety.py

tests/func/test\_add\_variety2.py

tests/func/test\_api\_exceptions.py tests/func/test\_unique\_id.py .

tests/unit/test\_cli.py

tests/unit/test\_task.py

-----------------coverage: platform darwin, python 3.6.2-final-0-------------------------

| Name | Stmts | Miss | Cover |

|----------------------------------------|-------|------|-------|

| src\tasks\\_\_\_\_init\_\_.py | 2 | 0 | 100% |

| src\tasks\api.py | 79 | 22 | 72% |

| src\tasks\cli.py | 45 | 14 | 69% |

| src\tasks\config.py | 18 | 12 | 33% |

| src\tasks\tasksdb\_pymongo.py | 74 | 74 | 0% |

| src\tasks\tasksdb\_tinydb.py | 32 | 4 | 88% |

| TOTAL | 250 | 126 | 50% |

========== 62 passed in 0.47 seconds ===============

8 слайд

Поскольку текущий каталог является tasks\_proj\_v2, а тестируемый исходный код находится в src, добавление опции --cov=src создает отчет о покрытии только для этого тестируемого каталога.

Как видите, некоторые файлы имеют довольно низкий и даже 0%, охват. Это полезные напоминания: tasksdb\_pymongo.py 0%, потому что мы отключили тестирование для MongoDB в этой версии. Некоторые из них довольно низкие. Проект, безусловно, должен будет поставить тесты для всех этих областей, прежде чем он будет готов к прайм-тайм.

Я полагаю, что несколько файлов имеют более высокий процент покрытия: api.py и tasksdb\_tinydb.ру . Давайте посмотрим на tasksdb\_tinydb.py и посмотрим, чего не хватает. Думаю, что лучший способ сделать это — использовать отчеты HTML.

Если вы снова запустите coverage.py с параметром --cov-report=html, будет создан отчет в формате HTML:

9 слайд

$ pytest --cov=src --cov-report=html

========= test session starts =======

plugins: mock-1.6.2, cov-2.5.1

collected 62 items

tests/func/test\_add.py

tests/func/test\_add\_variety.py tests/func/test\_add\_variety2.py tests/func/test\_api\_exceptions.py tests/func/test\_unique\_id.py .

tests/unit/test\_cli.py

tests/unit/test\_task.py

------------------coverage: platform darwin, python 3.6.2-final-0------------------------

Coverage HTML written to dir htmlcov

===================62 passed in 0.45 seconds======================

Затем можно открыть htmlcov/index.html в браузере, который показывает вывод на следующем экране:

Coverage report: 50%

| Module | Statements | Missing | Excluded | Coverage |

|------------------------------------|------------|---------|----------|----------|

| src\tasks\\_\_\_\_\_init\_\_.py | 2 | 0 | 0 | 100% |

| src\tasks\api.py | 76 | 22 | 0 | 72% |

| src\tasks\cli.py | 45 | 14 | 0 | 69% |

| src\tasks\config.py | 18 | 12 | 0 | 33% |

| src\tasks\tasksdb\_pymongo.py | 74 | 74 | 0 | 0% |

| src\tasks\tasksdb\_tinydb.py | 32 | 4 | 0 | 88% |

| Total | 250 | 126 | 0 | 50% |

coverage.py v4.5.2, created at 2019-02-21 23:02

10 слайд

Щелчок по tasksdb\_tinydb.py покажет отчет для одного файла. В верхней части отчета отображается процент покрытые строки, плюс сколько строк охвачено и сколько нет, как показано на следующем экране:

Coverage for src\tasks\tas X

Coverage for src\tasks\tasksdb\_tinydb.py: 88%

32 statements 28 run 4 missing o excluded

"""Database wrapper for TinyDB for tasks project."

import tinydb

Прокручивая вниз, вы можете увидеть пропущенные строки, как показано на следующем экране:

def list\_tasks(self, owner=None): # type (str) -> list[dict] """Return list of tasks."

if owner is None:

else:

return self.\_db.all()

return self.\_db.search(tinydb.Query().owner == owner)

def count(self): # type () -> int

"""Return number of tasks in db."""

return len(self.\_db)

def update(self, task\_id, task): # type (int, dict) -> () """Modify task in db with given task\_id."" self.\_db.update(task, eids=[task\_id])

def delete(self, task\_id): # type (int) -> () """Remove a task from db with given task\_id." self.\_db.remove(eids=[task\_id])

def delete\_all(self):

"""Remove all tasks from db.""" self.\_db.purge()

def unique\_id(self): # type () -> int

"""Return an integer that does not exist in the db.""" i = 1

while self.\_db.contains(eids=[i]):

i += 1

return i

def stop\_tasks\_db(self):

"""Disconnect from DB."

Pass

**Техническое задание (ТЗ) для продолжения диалога**

**Контекст:**

* Мы работаем над заданием по тестированию асинхронных методов и метрик покрытия кода.
* Максимальное количество баллов: 10.
* Крайний срок сдачи: 14.11.2024.

**Шаги, которые были выполнены:**

1. **Шаг 1: Установка WSL и Ubuntu.**
   * Открыть PowerShell от имени администратора.
   * Ввести команду wsl --install .
   * Перезагрузить компьютер (необходимо для завершения установки).

**Следующий шаг:**

* После перезагрузки продолжить с установки необходимых инструментов для тестирования, таких как pytest и pytest-cov .