BO·TT·EGA
IT minds

Program szkolenia:

Asyncio

Informacje:

Nazwa:	**Asyncio**
Kod:	**python-asyncio**
Kategoria:	Python
Odbiorcy:	developerzy, architekci
Czas trwania:	3 dni
Forma:	30% wykłady / 70% ćwiczenia

Szkolenie z pisania asynchronicznego w Pythonie z wykorzystaniem asyncio. Oparte o dwa studia przypadku - czat oraz giełdę.

Dzień 1: Wstęp, kiedy asyncio to właściwe podejście, projektowanie programów z użyciem asyncio,

Dzień 2: Wzorce w asynchronicznym programowaniu, używanie kodu niekompatybilnego z asyncio

Dzień 3: Testowanie, profilowanie i debuggowanie kodu w asyncio, przegląd podobnych rozwiązań

Zalety szkolenia:

- Wzorce
- Narzędzia
- Zrozumienie pryncypiów

BOTTEGA IT minds

Szczegółowy program:

1. Problem C10k ze studium przypadku - piszemy czat

- 1.1. Podejście z użyciem puli procesów
- 1.2. Podejście z użyciem puli wątków
- 1.3. Współprogramy (ang. coroutines)

2. Coroutines w praktyce

- 2.1. Jawne oddanie kontroli async/await
- 2.2. Dlaczego czasem asyncio.sleep(0) to dobry pomysł
- 2.3. Czekanie na wiele zadań naraz - gather/wait
- 2.4. Uruchamianie w tle przy pomocy Task / asyncio.ensure_future

3. Pętla zdarzeń

- 3.1. Jedyna zasada: Nie blokuj nigdy pętli zdarzeń!
- 3.2. Zasada działania
- 3.3. Alternatywne implementacje
  - 3.3.1. uvloop
- 3.4. Monitorowanie ilości współprogramów
- 3.5. Uruchamianie kilku pętli zdarzeń

4. Kiedy asyncio to dobry wybór

- 4.1. Problemy CPU-bound
- 4.2. Problemy IO-bound

5. Wzorce asynchronicznego przetwarzania w asyncio

- 5.1. Locki
- 5.2. Timeouty
- 5.3. Wywołania zwrotne (ang. callbacks)

BOTTEGA IT minds

- 5.4. Komunikacja przez współdzieloną pamięć i dlaczego to zły pomysł
- 5.5. Komunikacja przez kolejki
- 5.6. Workery
- 5.7. Kontrola przepływu (ang. flow control), back-pressure

6. Wbudowane elementy do szybszego budowanie aplikacji

- 6.1. Osadzamy serwer HTTP w aplikacji czatu do serwowania statystyk
  - 6.1.1. Strumienie (ang. streams)
  - 6.1.2. Protokoły i Transporty

7. Używanie niekompatybilnych bibliotek i kodu CPU-bound z asyncio

- 7.1. asyncio.to_thread
- 7.2. Executor
  - 7.2.1. oparty o wątki
  - 7.2.2. oparty o procesy

8. Uruchamianie kodu CPU-bound w asyncio

- 8.1. Przez Executor oparty o procesy
- 8.2. Przez kolejkę zadań

9. Testowanie kodu asyncio

- 9.1. podstawa izolacji - tworzenie nowej pętli zdarzeń dla każdego testu
- 9.2. pytest-asyncio
- 9.3. AsyncMock

10. Profilowanie i monitorowanie rozwiązań w asyncio

- 10.1. cProfile
- 10.2. yappi
- 10.3. pyinstrument
- 10.4. Użycie zewnętrznych usług klasy APM

BO·TT·EGA
IT minds

11. Debuggowanie rozwiązań w asyncio

11.1. Włączanie trybu debuggowania

12. Przegląd innych podejść do asynchronicznego przetwarzania w Pythonie

12.1. Celery i podobne kolejki zadań

12.2. Trio

12.3. Twisted

12.4. Gevent