

Zadania rekrutacyjne PsycheON

Wymagania ogólne:

- Rozwiązania powinny działać w Pythonie 3.X, w środowisku Ubuntu ≥ 18.04 .
- Dozwolona jest instalacja bibliotek pythonowych przez odpowiednie *requirements* w *setup.py*.
- Niedozwolona jest konieczność instalowania bibliotek systemowych przy pomocy apt, curl i pochodnych.
- Kod powinien być dobrze udokumentowany oraz czytelny.

Zadanie 1

Kod aplikacji zapisz w katalogu **todolist**.

Napisz prostą aplikację internetową „lista todo” we frameworku Django z wykorzystaniem bazy sqlite3, która:

- a) Zawiera model z następującymi parametrami: id (int, identyfikator), title (str, tytuł zadania), done (bool, czy zadanie zostało zrobione), author_ip (str, adres ip osoby tworzącej), created_date (datetime, data utworzenia), done_date (datetime, data zakończenia zadania).
- b) Posiada widok dostępny pod adresem `‘/todolist’` z metodą GET, który w postaci JSONA zwraca listę wcześniej zapisanych w bazie sqlite zaplanowanych zadań.
- c) Posiada widok `‘/todolist’` z metodą POST, który przyjmuje na wejściu JSON z parametrami: title, done, done_date. Widok ma dodać zadanie o tytule w podanym polu title (jedyne wymagane parametry). Oznaczyć jako zrobione lub niezrobione zależnie od wartości pola done, dodatkowo ustawić adres IP requesta jako author_ip. W przypadku niepodania pola done, należy ustawić zadanie jako niewykonane. Dodatkowo w przypadku podania done: true oraz done_date, należy ustawić obie wartości w bazie, natomiast w przypadku podania done: true oraz niepodania done_date, należy użyć bieżącego czasu jako done_date. Ostatnim warunkiem, to zwrócenie kodu HTTP 400 w przypadku podania done: false oraz done_date innego niż null.
- d) Posiada widok `‘/todolist/<id_zadania>’` z metodą GET, który w przypadku podania identyfikatora zadania, które nie istnieje, zwraca HTTP 404, natomiast w przypadku istniejącego zadania, powinien być zwrócony kompletny obiekt zadania.
- e) Posiada widok `‘/todolist/<id_zadania>’` z metodą DELETE, który w przypadku podania identyfikatora zadania, które nie istnieje, zwraca HTTP 404, lecz w przypadku istniejącego zadania, to zadanie zostanie usunięte z bazy danych.
- f) Posiada testy, które sprawdzą poprawność działania programu.

Zadanie 2

Kod zadania zapisz w pliku cars.py.

Napisz klasę Car, która będzie reprezentowała samochód. W momencie tworzenia instancji tej klasy, mają być możliwe do podania cztery parametry: brand (marka samochodu), tank_capacity (maksymalna objętość baku w litrach), tanked_fuel (liczba litrów paliwa w baku) oraz engine_type (rodzaj silnika Diesel/Gasoline). Podczas tworzenia instancji należy zalogować na poziomie INFO komunikat:

New car of brand <brand>, with tank full in XXX%.

Gdzie <brand> ma być marką samochodu, natomiast XXX procentowym napełnieniem baku z dokładnością do 1 miejsca po przecinku. Obiekty klasy Car muszą być wyposażone w następujące metody:

- a) fill_tank, która bez podania argumentów napełni bak samochodu do pełna. Metoda ma zwracać liczbę litrów dolanego paliwa.
- b) fill_tank, która z podaniem argumentu liters wypełnia bak podaną liczbą litrów paliwa. W przypadku, gdyby maksymalna pojemność baku miała być przekroczona, oczekiwane jest rzucenie odpowiedniego wyjątku z komunikatem. Metoda ma zwracać liczbę litrów dolanego paliwa.
- c) fill_tank, która w przypadku napełniania baku samochodu z silnikiem Diesla uniemożliwi to oraz wyrzuci wyjątek EnvironmentalError z komunikatem: ON fuel not available, because of environmental restrictions. Change engine as soon, as possible.
- d) empty_tank, która po podaniu parametru limit, opróżni bak samochodu do podanego limitu, gdzie limit jest wartością z zakresu <0;1> i reprezentuje procentowe napełnienie zbiornika. Dla przykładu, dla samochodu ze 100 litrowym bakiem napełnionym do pełna, podanie parametru limit o wartości 0.4, opróżni bak o 60 litrów. Metoda ma zwracać liczbę opróżnionych litrów.
- e) Metodę, która po wpisaniu do konsoli pythona nazwy obiektu i naciśnięciu klawisza enter pokaże: <Car object at <memory_addres> registered under the <brand_name> brand, with tank full in XXX%> (XXX% - procentowe zapelnienie baku).

Metody muszą być zabezpieczone przed podaniem niepoprawnych danych. Przykładowo niemożliwe powinno być wlanie 100 litrów paliwa do baku o pojemności 50 litrów itd.