**Zadanie na zaliczenie(TODO List)**

Bazując na zdobytej wiedzy z zajęć, stwórz aplikację **KONSOLOWĄ** zarządzającą listą zadań. Klasyczna aplikacja TODO list.

Poglądowo przykład takiej aplikacji znajduje się pod adresem: <http://todomvc.com/examples/vue/>

Aplikacja konsolowa, czyli taka która będzie bazowała na argumentach wejściowych(node app.js someCommand --someArg '...').

**Ogólny zarys aplikacji**

1. Aplikacja powinna mieć możliwość wyświetlania listy, dodania oraz usunięcia rekordu z naszej listy. Dane powinny być zapisane do pliku. **(format jak i sposób przetrzymywania danych zostawiam programiście)**
2. Dodatkowo, aplikacja powinna pozwalać na zmianę statusu wykonanego zadania. (warto pomyśleć o nadawaniu jakiegoś identyfikatora)
3. Dodając opcję z pkt. 2 powinniśmy dodać możliwość filtrowania naszej listy po statusie(wszystkie/aktywne/kompletne).
4. Aplikacja powinna pozwalać na pogrupowanie naszej listy na odpowiednie grupy, np.:

W naszej liście do zrobienia dodajemy pozycje:

Napisać maila do Jan Nowak

Kupić 6 jaj

Kupić mleko

...

W takim przypadku dodając elementy związane z zakupami powinniśmy nadać jakąś grupę(**nie bawimy się tutaj w żadne relacje itp. PROSTA STRUKTURA DANYCH!**).

Dzięki dodaniu grupowaniu po określonej kategorii powinniśmy dodać nowe filtrowanie do wyświetlania naszej listy.

1. (\*) Do przećwiczenia odbieranie oraz wysyłanie danych do serwera.

Aplikacja powinna pozwalać na synchronizację danych. Dodatkowa komenda pozwalająca na manualne pobieranie oraz wysyłanie danych do serwera, np.: node app.js upload/download / itp...

Adres do api: [http://api.quuu.linuxpl.eu/todo/{kod\_autoryzacyjny}](http://api.quuu.linuxpl.eu/todo/%7Bkod_autoryzacyjny%7D)

np.: <http://api.quuu.linuxpl.eu/todo/12345678>

Każdy z uczestników studiów powinien otrzymać unikatowy kod autoryzacyjny na swoją pocztę(jeżeli tak się nie stało, proszę o kontakt!).

Wystawione API jest typu REST. Aby pobrać dane wysyłamy żądanie metodą GET na adres z poprawnym kodem. Aby zaktualizować dane na serwerze wysyłamy żądanie metodą POST lub PUT na ten sam adres.

**UWAGA!** API przyjmie jedynie poprawny obiekt JSON oraz maksymalną długość znaków do 500000. Jeżeli będzie błąd zostanie zwrócony odpowiedni statusCode(400/404/500).

**Ważne!**

Wygląd jak i działanie aplikacji, jak również dobór bibliotek zewnętrznych zależy od programisty! Liczę na pomysłowość i różne podejścia do wykonania zadania!

**Rozwiązania w postaci spakowanej lub jako link do repozytorium GIT proszę przesłać do 07.02.2019 w wiadomości zwrotnej w której został przekazany indywidualny klucz dostępu do API.**

**POWODZENIA! ;)**