

Node.js

Laboratorium 2

1. Stworzyć aplikację która pozbędzie się zduplikowanych danych w tablicy

Dane w tablicy:

```
['ala', 3, 'ma', 'kota', 1, 'ala', 1, 4, 3]
```

W konsoli powinna zostać wyświetlona tablica bez zduplikowanych wartości:

```
['ala', 3, 'ma', 'kota', 1, 4]
```

W tym zadaniu wykorzystać zewnętrzną bibliotekę 'lodash'

(<https://www.npmjs.com/package/lodash>)

2. Zainstalować globalny moduł 'nodemon' (<https://www.npmjs.com/package/nodemon>), aby możliwe było automatyczne restartowanie w przypadku wykrycie zmian napisanej przez nas aplikacji.

3. Stworzyć aplikację która porówna dwie tablice i wyświetli w konsoli różnicę między nimi.

```
const tabA = ['ala', 'ma', 'kota']  
const tabB = ['ala', 'ma', 'psa']
```

Wykorzystać zainstalowaną bibliotekę 'lodash'.

4. Wykorzystując bibliotekę 'lodash' znaleźć minimalną oraz maksymalną wartość w tablicy:

```
[3, 5, -20, -1002, 234, 542, 6, 23, -3, 8]
```

5. Wykorzystując zdobytą wiedzę z poprzednich zajęć skorzystaj z zewnętrznej biblioteki 'yargs' (<https://www.npmjs.com/package/yargs>) i stwórz aplikację mnożenia i dzielenia wprowadzonych argumentów

```
> node app.js --a=5 --b=7 --operation=*
```

Konsola powinna zwrócić wynik operacji mnożenia...

6. Stworzyć aplikację która wyświetli na ekranie przywitanie użytkownika po 5 sekundach od uruchomienia aplikacji.

Wykorzystaj tutaj wiedzę z poprzednich zajęć (podpowiedź: moduł 'os', funkcja userInfo()) oraz funkcję(setTimeout()) która pozwala na wrzucenie naszej funkcji do node api.

7. Zmodyfikuj aplikację z pkt. 5 tak aby operacje mnożenia i dzielenia były asynchroniczne(callback function)

Funkcja mnożenia i dzielenia powinna przyjmować 3 argumenty 'a', 'b' oraz 'callback'

Przekazana funkcja ma za zadanie wyświetlić wynik naszego działania;

Np.:

```
myMath.divide(3, 5, function (result) {  
    console.log(result);  
});
```

8. Stwórz aplikację która pobierze informacje o naszym koncie z API Github'a.

Adres do pobrania danych:

https://api.github.com/users/{nazwa_uzytkownika}

np.:

<https://api.github.com/users/piotar>

W tym zadaniu wykorzystamy zewnętrzny moduł 'request'

(<https://www.npmjs.com/package/request>). Pakiet ten udostępnia nam funkcje pozwalające na wysłanie żądania do zewnętrznego serwera i pobranie danych.

9. Rozszerzmy aplikację z pkt.8 o zapis asynchroniczny naszego użytkownika do pliku i wyświetlmy komunikat o poprawnej operacji.

10. Rozszerzmy aplikację z pkt.8 o wprowadzanie loginu poprzez parametr, np.:

```
> node app.js --login=piotar
```

11. Stwórz aplikację która pobierze dane dotyczące pogody i wyświetl ją na konsoli.

Adres do pobrania danych:

<https://api.openweathermap.org/data/2.5/weather?q=Białystok&APPID=0ed761300a2725ca778c07831ae64d6e>

W tym zadaniu wykorzystamy zewnętrzny moduł 'request'

(<https://www.npmjs.com/package/request>). Pakiet ten udostępnia nam funkcje pozwalające na wysłanie żądania do zewnętrznego serwera i pobranie danych.