

## Contents

<b>1</b>	<b>Projekt 6: Serwis</b>	<b>1</b>
1.1	Wprowadzenie . . . . .	1
1.2	Ocenianie . . . . .	1
1.3	Szablon: serwer . . . . .	2
1.4	Szablon: testy . . . . .	3
1.5	Zasada DRY . . . . .	3

## 1 Projekt 6: Serwis

### 1.1 Wprowadzenie

Postaw serwis na Heroku [<https://www.heroku.com/>], gdzie udostępnisz użytkownikowi wybrane przez siebie dane z portalu danepubliczne.gov.pl w formie określonej w szablonie. Wykorzystaj MongoDB do magazynowania swoich danych. Zaprojektuj odpowiednie testy (patrz szablon). Termin oddawania to 21.01.2018



```
{
  "_id": "5a4bbe3aa7caf443b2910f94",
  "__v": 0,
  "data": "2017-11-01 00:00",
  "03": 25.54,
  "modifiedAt": null,
  "modified": false
},
{
  "_id": "5a4bbe3aa7caf443b2910f95",
  "__v": 0,
  "data": "2017-11-01 01:00",
  "03": 25.85,
  "modifiedAt": null,
  "modified": false
},
{
  "_id": "5a4bbe3aa7caf443b2910f96",
  "__v": 0,
  "data": "2017-11-01 02:00",
  "03": 26.46,
  "modifiedAt": null,
  "modified": false
},
{
  "_id": "5a4bbe3aa7caf443b2910f97",
  "__v": 0,
  "data": "2017-11-01 03:00",
  "03": 15.11,
  "modifiedAt": null,
  "modified": false
},
}
```

### 1.2 Ocenianie

Wykonanie poprawnie tego zadania jest na ocenę 4.0. Aby uzyskać ocenę wyższą należy wyróżnić się spośród grupy kreatywnością. Składa się na to przede wszystkim:

1. Oryginalne rozwiązanie techniczne
2. Nawiązanie konstruktywnej (korzystnej dla obu stron) interakcji z Ministerstwem Cyfryzacji lub innym podmiotem publicznym związanym z portalem danepubliczne.gov.pl w ramach zagadnień dotyczących projektu (proszę mnie dołączyć do korespondencji w CC).

### 1.3 Szablon: serwer

```
const express = require('express');
const {mongoose} = require('./db/mongoose.js');
const {Ozon} = require('./models/ozon.js');

let app = express();

app.get('/', (req, res) => {
  res.send("<h1>Stężenia ozonu w Brennej</h1>
  <p>Stacja znajduje się na stoku Starego Gronia
  w Brennej i jest jednym z elementów Multipoligonu
  Zintegrowanego Monitoringu Środowiska Instytutu
  Ekologii Terenów Przemysłowych</p>");
});

app.post('/ozon', (req,res) => {
});

app.get('/ozon', (req,res) => {
});

app.get('/ozon/:id', (req, res) => {
});

app.delete('/ozon/:id', (req, res) => {
});

app.patch('/ozon/:id', (req,res) => {
});

app.listen(port, () => {
  console.log(`server is running on port ${port}`);
});
```

## 1.4 Szablon: testy

```
describe('POST /ozon', () => {
  test('powinno zapisać do bazy danych dokument', callback)
  test('powinno nie zapisać do bazy danych dokument', callback)
}
describe('GET /ozon', () => {
  test('powinno zwrócić listę dokumnetów z bazy danych', callback)
}
describe('GET /ozon/:id', () => {
  test('powinno zwrócić dokument o szukanym id', callback)
  test('powinno zwrócić kod 404 gdy id niepoprawne', callback)
  test('powinno zwrócić kod 404 gdy id nie ma w bazie danych', callback)
}
describe('DELETE /ozon/:id', () => {
  test('powinno usunąć dokument o szukanym id', callback)
  test('powinno zwrócić kod 404 gdy id niepoprawne', callback)
  test('powinno zwrócić kod 404 gdy id nie ma w bazie danych', callback)
}
describe('PATCH /ozon/:id', () => {
  test('powinno uaktualnić data w dokumencie o szukanym id', callback)
  test('powinno usunąć wartość z modifiedAt gdy modified false', callback)
  test('powinno zwrócić kod 404 gdy id niepoprawne', callback)
}
```

## 1.5 Zasada DRY

Kod ma być napisany z zastosowaniem zasady DRY (don't repeat yourself). <https://pl.wikipedia.org/wiki/DRY> Jeśli fragment kodu się powtarza to umieścić go w funkcji, obiekcie, pliku itp.