

# Tablica informacyjna

oparta na komputerze jednopłytkowym Raspberry Pi

Mateusz Norel (126901)  
Tomasz Walczak (126851)  
Mateusz Grabuszyński (126917)  
Informatyka WE, TI L-2

# Serwer (1)

Dokonano zakupu serwera VPS celem:

- postawienia serwera WWW dla managera oraz API
- przetrzymywania bazy danych MongoDB zawierającej wpisy serwowane przez API

System operacyjny: Ubuntu Server 16.04 LTS 64bit

CPU: 1

RAM: 1 GB

SSD: 20 GB

Transfer miesięczny: 2 TB

Dostawca: ArubaCloud

Lokalizacja: Czechy

# Serwer (2)

Na VPSie postawiono bazę dokumentową MongoDB. W bazie znajdują się elementy, które zwracane są przez RESTowe API do wyświetlenia w rotatorach:

**GET /api** – zwraca dokument JSON z danymi rotatorów.



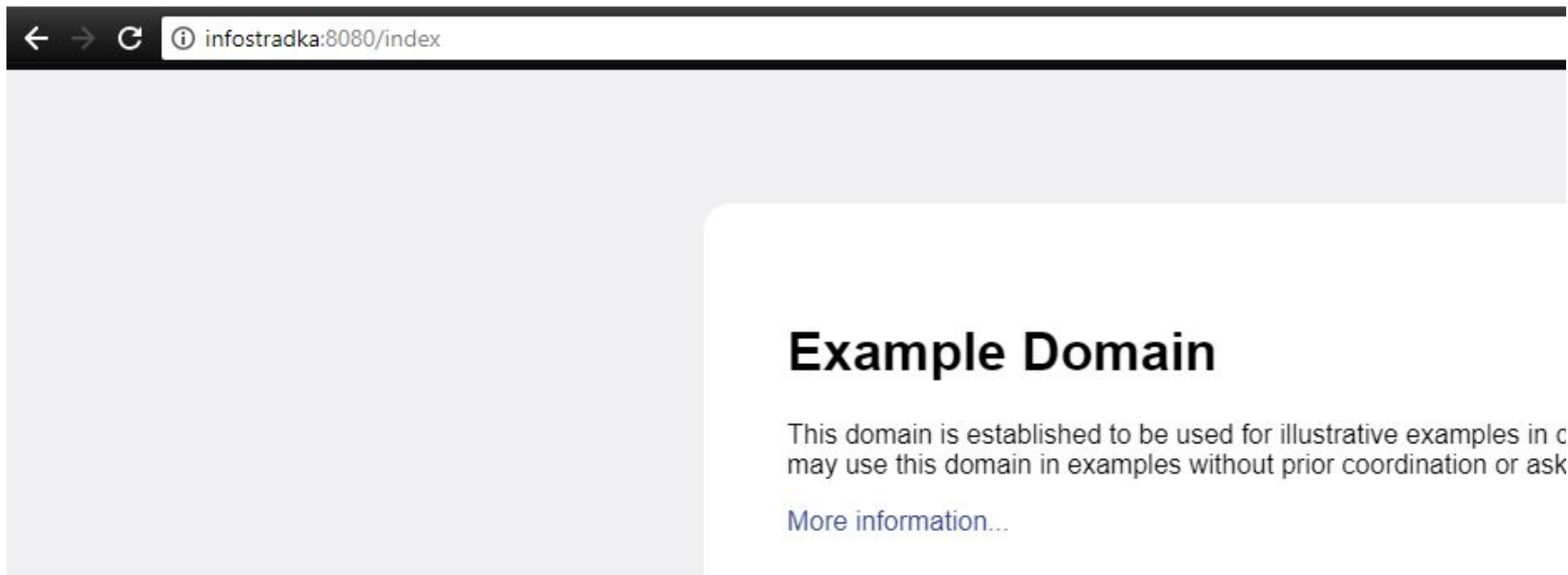
infostradka:8080/api

```
{
  "left": [
    {
      "content": {
        "source": "http://distribution.bbb3d.renderfarming.net/video/mp4/bbb_sunflower_1080p_60fps_normal.mp4",
        "subtitles": ""
      },
      "duration": 5.0,
      "since": "2018-04-10 22:00",
      "type": "video",
      "until": "2019-04-10 22:00"
    }
  ],
  "news": [
    {
      "content": "...szarobure obydwu czy co\u015b tam jako\u015b tam co\u015b tam.<br><br>BARDZO DU\u017bO TEKSTU TUTAJ...",
      "duration": 15.0,
      "important": 0.0,
      "since": "2018-04-10 22:00"
    }
  ]
}
```

# Serwer (3)

Napisano od nowa bądź poprawiono wcześniej napisaną logikę dla następujących endpointów:

**GET /index** – zwraca szablon strony WWW wyświetlanej na ekranach (`render_template('index.html')`);



# Serwer (4)

**GET /manager** – zwraca szablon strony WWW managera (`render_template('manager.html')`).



**Witaj !**

Liczba podłączonych urządzeń:  
Liczba wyświetlanych elementów:

# Klient

Napisano obsługę dat ważności wyświetlania elementów w rotatorach (pola “since” oraz “until”).

Jeśli element do wyświetlenia jest przeterminowany rotator przechodzi do kolejnego.

Rozważamy, by zliczać przeterminowane elementy i informować o nich administratora systemu, wysyłając ostrzeżenie odpowiedniego poziomu.

# Github

Poprawiono opis README.md w repozytorium tak, aby odpowiadał dzisiejszemu stanowi. Opisano proces instalacji odpowiednich paczek na serwerze, a także uruchomienie i obsługę serwisów.

## Serwer

Instalacja wersji na dzień dzisiejszy (bez screenshotów, obsługi wideo i innych):

1. Należy pobrać kod z repozytorium przy użyciu komendy `git clone https://github.com/MateuszGrabuszynski/ict-infostradka.git` ;
2. Przejść do folderu infostradka-server ( `cd infostradka-server` );
3. Zainstalować odpowiednie biblioteki poleceniem `pip3 install -r requirements.txt` ;
4. Uruchomić serwer `python3 -m swagger_server` . Mimo wyłączonego trybu debug w konsoli mogą być wypisywane fragmenty zapytań bądź inne niezbędne przy wcześniejszym developmencie elementy. Ostateczna wersja nie będzie zawierała takich elementów.

# Co dalej?

- Dokończenie managera (serwer, html+JS);
- Dodawanie zrzutów ekranu (klient, Bash + poprawki serwera);
- Klucze sesyjne i hasze klientów (serwer + klient, JS);
- Odporność na brak sieci/brak danych (klient, JS);
- Dodanie obrazu \*.img oprogramowania zainstalowanego na Raspberry (klient, GitHub);
- Refaktoryzacja kodu rotatorów (klient, JS).