Sensor Dummy service

SensorDummy service je napisan u Node.js-u. Koristi dodatne npm pakete za čitanje iz CSV fajla i za konektovanje i rad nad MQTT brokerom. Prilkom pisanja docker compose file-a, mora se voditi računa da ovaj kontejner krene za izvršavanjem tek nakon što se EMQX broker startuje. On podatke pročitane iz .csv fajla pablišuje na topic sensor_dummy/values.

Analytics service

Analytics service je napisan u .Net-u. Koristi dodatne pakete za rad sa MQTT-om i sa InfluxDB-jem. To su paketi MQTTnet i InfluxDB.Client. Čita podatke sa sensor_dummy/values topic-a, zatim ih pablišuje na analytics/values topic eKuiper-u. Obrađene podatke eKuiper-a prima sa eKuiper/anomalies topic-a, i upisuje ih u InfluxDB.

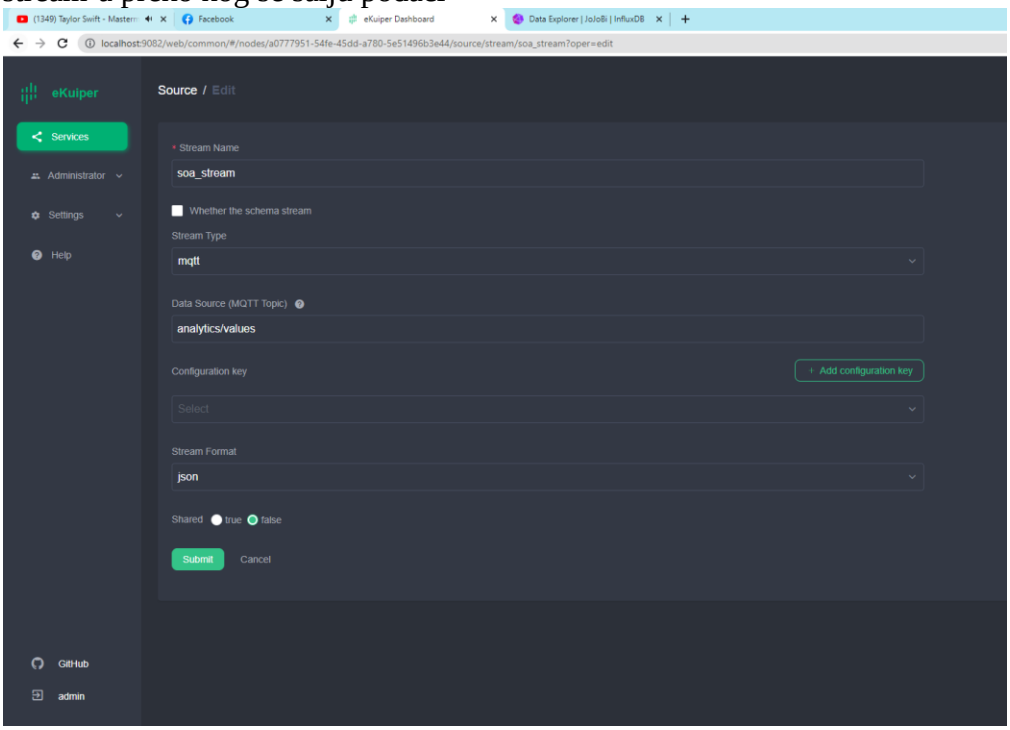
MQTT broker

Koriscen je EMQX mqtt broker

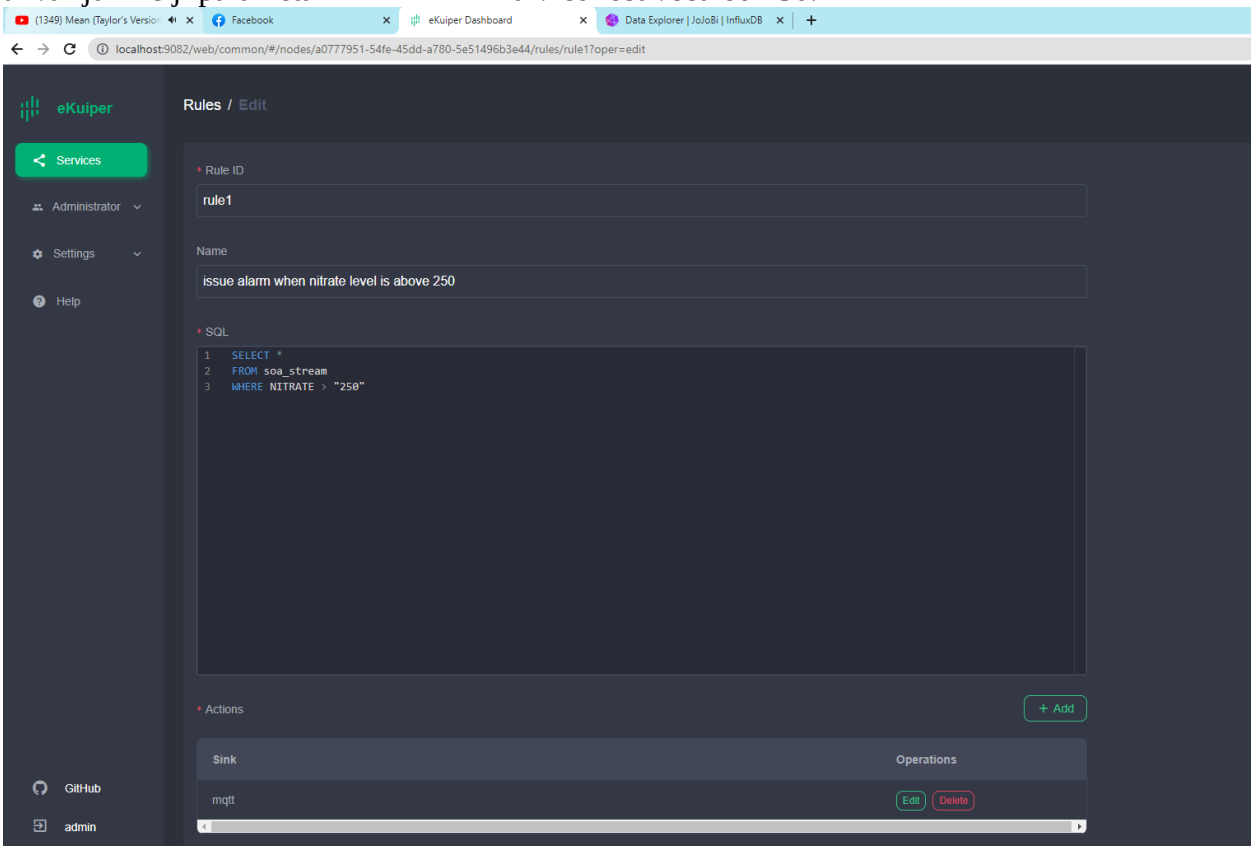
eKuiper service

U docker-compose fajl je potrebno za MQTT_SOURCE__DEFAULT__SERVER navesti adresu računara, ni sa jednom drugom adresom se neće zakačiti na mqtt broker. Uz pomoc localhost:9083 pokrecemo manager za eKuiper I koriscenjem interfejsa kreiramo *stream*, a zatim I pravilo- *Rule*, uz pomoc kojeg se vrsi filtriranje podataka. Logovanje se vrsi unosom *admin* za username I *public* za password.

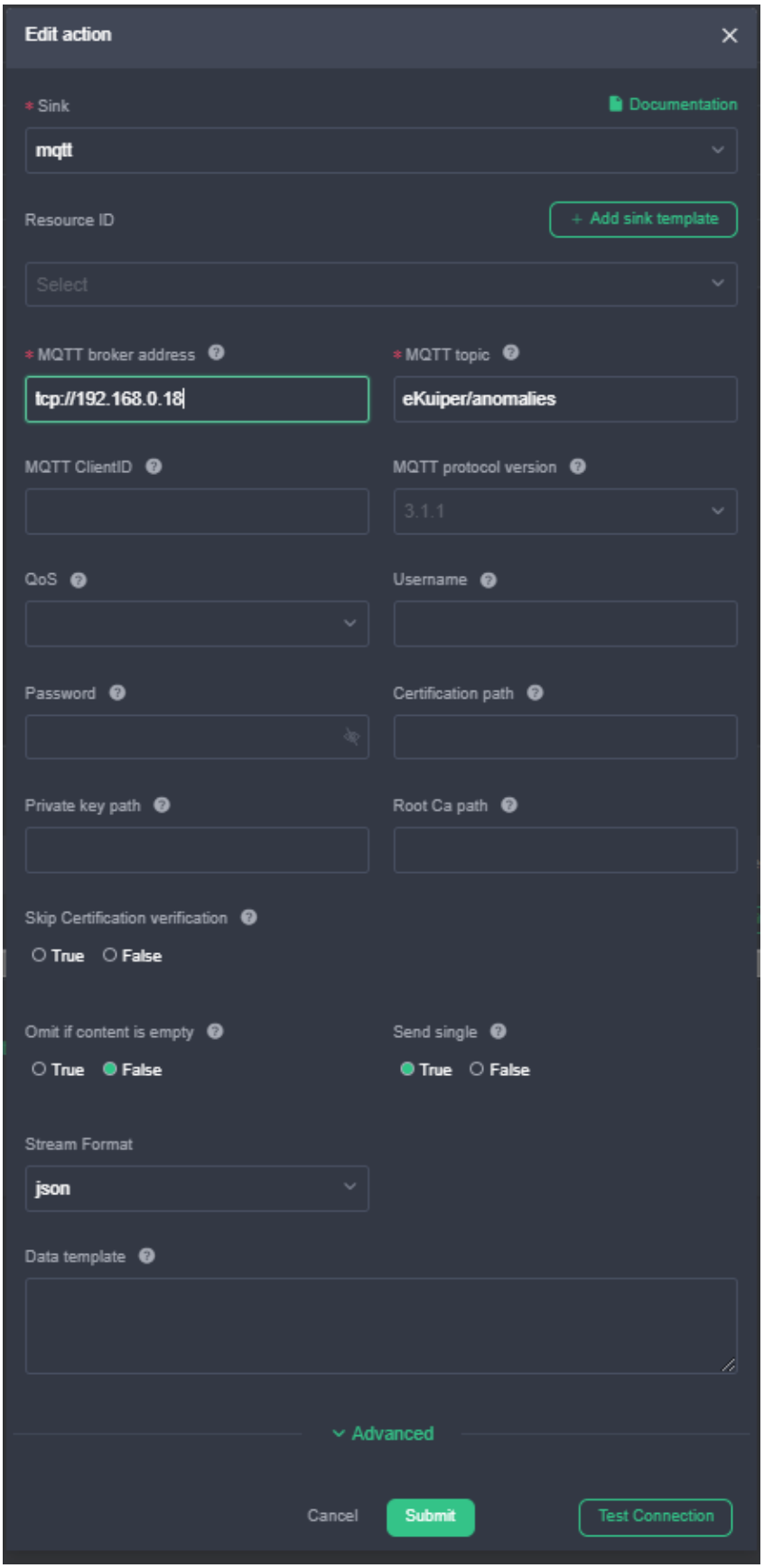
Klikom na *stream* karticu, unosom podataka sa slike I na kraju klikom na *Stream* dugme vrsi se kreiranje stream-a preko kog se salju podaci



Dalje je potrebno otici na *Rules* karticu I kreirati pravilo po kome se filtriraju podaci iz prethodno kreiranog stream-a. Uneti podatke sa slike. U nasem projektu se detektuju kao anomalija akvarijumi ciji parametar NITRATE ima vrednost vecu od 250.

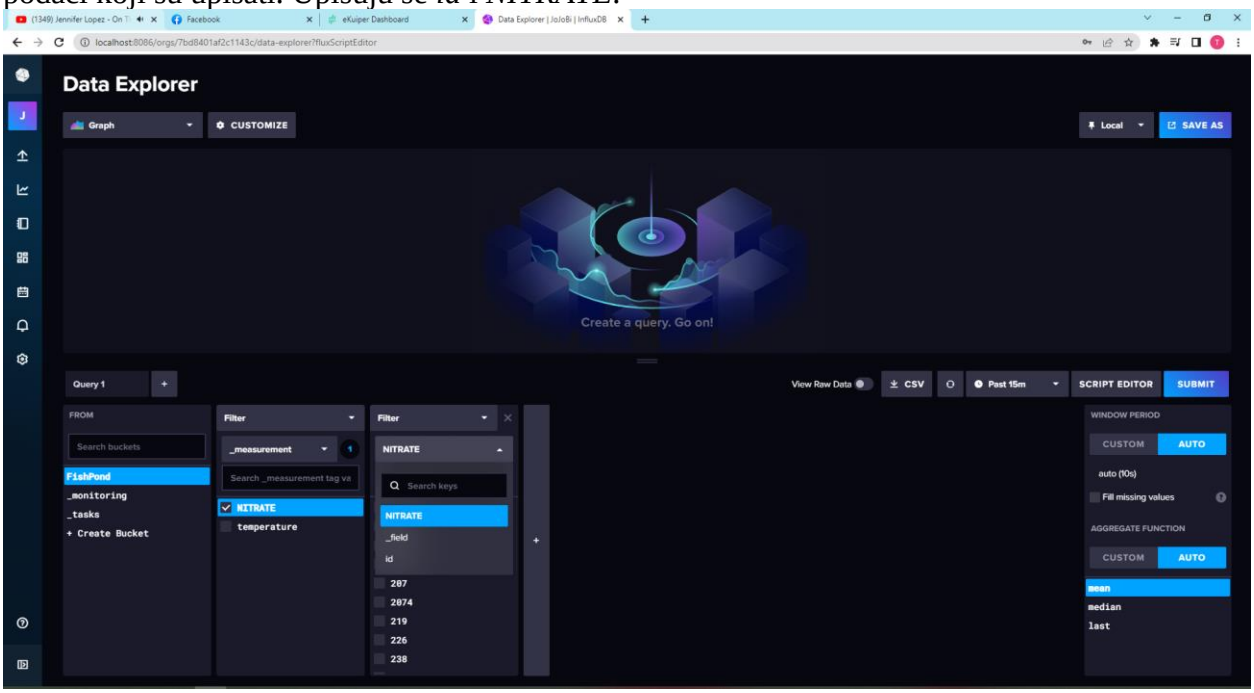


Potrebno je zatim kreirati akciju klikom na *add* dugme u okviru odeljka *Actions*, popuniti polja sledecim podacima. Na kraju je potrebno kliknuti *Submit* i startovati Rule.


Edit action
* Sink
Documentation
mqtt
Resource ID
+ Add sink template
Select
* MQTT broker address
tcp://192.168.0.18
* MQTT topic
eKuiper/anomalies
MQTT ClientID
MQTT protocol version
3.1.1
QoS
Username
Password
Certification path
Private key path
Root Ca path
Skip Certification verification
True
False
Omit if content is empty
True
False
Send single
True
False
Stream Format
json
Data template
Advanced
Cancel
Submit
Test Connection

InfluxDB

Potrebno je logovati se kreitanjem naloga. Prilikom kreiranja naloga smo kreirale i bucket *FishPond* u koji ce se smestati podaci procitani sa *eKuiper/anomalies* topic-a. Nalog je kreiran sa podacima: username-*bisenicct*, password-*Tijana22!*, organization-*JoJoBi*. Na slici ispod se vide podaci koji su upisati. Upisuju se *id* i *NITRATE*.



AsyncAPI

Kako u projektu postoje 2 servisa, kreiran je po jedan *yaml* fajl za svaki.
Oba fajla imaju iste postavke, prikazane na slici ispod.

```yaml
asyncapi: '2.6.0'
info:
  title: SensorDummy
  version: 1.0.0
  description: This service is in charge of reading data from fish pond
  license:
    name: Apache 2.0
    url: 'https://www.apache.org/licenses/LICENSE-2.0'
servers:
  mosquitto:
    url: mqtt://localhost
    protocol: mqtt
```

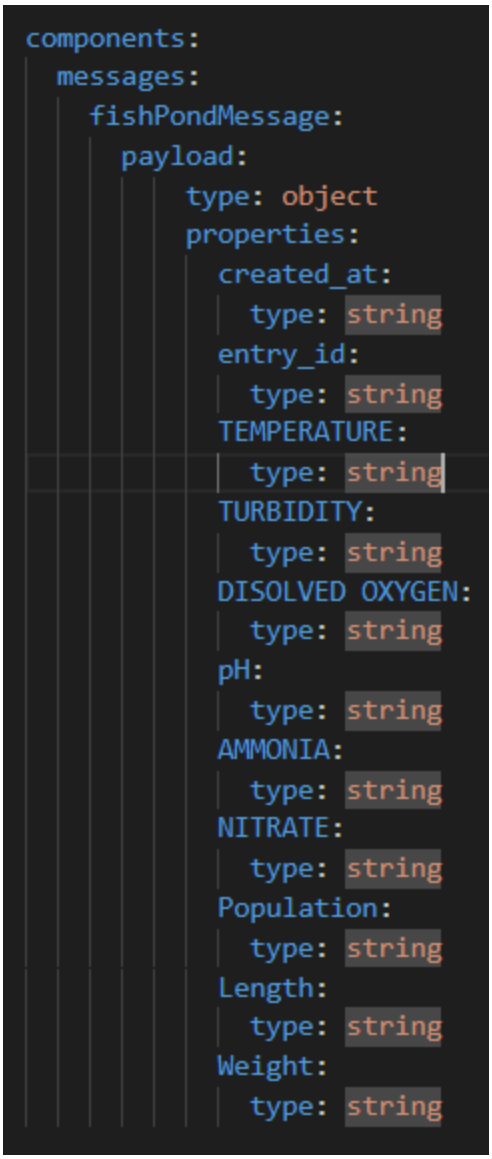
Razlika je u *title* i *description* koji se popunjavaju u skaldu sa nazivom i funkcionalnostima sistema.

Channels komponenta definise sve topic-e na koje je servis postavlja, odnosno sa kojih servis prima podatke. U nastavku je slika na kojoj vidimo topic-e na koje je analytics servis subscribe-ovan, odnosno one na kojima postavlja podatke.

```yaml
channels:
  sensor_dummy/values:
    publish:
      summary: Receive data from dummy
      operationId: PublishAsync
      message:
        #name: fishPondMessage #Dummy
        $ref: '#/components/messages/fishPondMessage'
  analytics/values:
    subscribe:
      summary: Publish data to check for anomalies
      operationId: ApplicationMessageReceivedAsync
      message:
        #name: fishPondMessage #Analytics
        $ref: '#/components/messages/fishPondMessage'
  eKuiper/anomalies:
    publish:
      summary: Recv data from ekuiper
      operationId: WriteToDatabase
      message:
        #name: fishPondMessage #Analytics
        $ref: '#/components/messages/fishPondMessage'
```

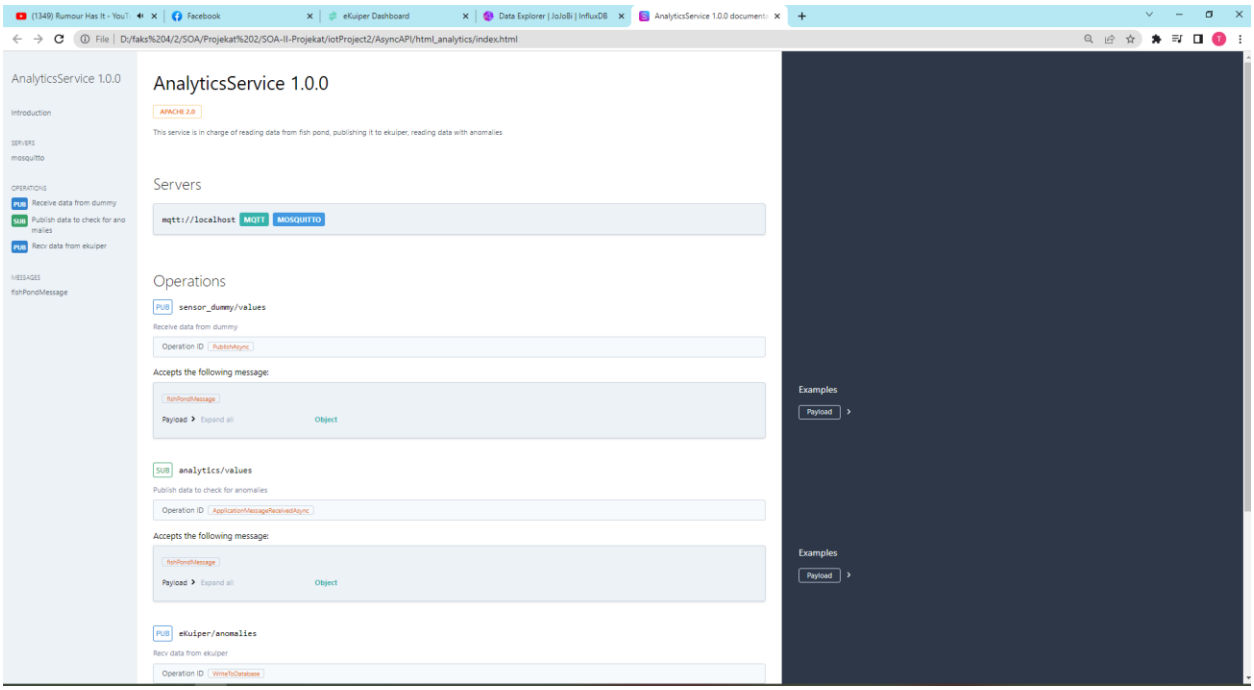
Prvo se navodi naziv topica, zatim se sa *publish/subscribe* definise da li se sa tog topika primaju, odnosno da li se na njega postavljaju podaci. *OperationId* predstavlja funkciju kojom se okida operacija citanja, odnosno postavljanja podataka na topic, i na kraju polje *message* predstavlja opis strukture poruke koja se salje, odnosno prima. Polje *name* nije obavezno, te je u ovom konkretnom slucaju zakomentarisano.

Kako se u nasem projektu uvek salju svi podaci procitani iz baze, struktura poruke je u svakom slucaju sledeca:



Struktura *sensor_dummy.yaml* fajla je identicna.

Izvrsenjem komande „ag ./**asyncapi.yaml** @asyncapi/html-template -o **html“**, gde umesto *asyncapi.yaml* navodimo ime naseg yaml fajla, a umesto *html* navodimo ime foldera u kojem zelimo da se kreira stranica. Pokretanjem *index.html* otvara se generisana dokumentacija.



Izvrsavanjem komande „ag ./**asyncapi.yaml** @asyncapi/nodejs-template -p server=mosquitto -o **example“** kreiramo node aplikaciju cijim pokretanjem mozemo testirati da li nasa aplikacija ispravno radi. Umesto *asyncapi.yaml* navodimo odgovarajucu yaml specifikaciju, a umesto *example* navodimo folder u kome zelimo da se apliakcija kreira. Komandom npm start pokrecemo aplikaciju i dobijamo sledeci izlaz:

```
D:\faks 4\2\SOA\Projekat 2\SOA-II-Projekat\iotProject2\AsyncAPI\dummy>npm start

> sensor-dummy@1.0.0 start
> node src/api/index.js

 PUB  Will eventually publish to sensor_dummy/values
SensorDummy 1.0.0 is ready!

   MQTT adapter is connected!

D:\faks 4\2\SOA\Projekat 2\SOA-II-Projekat\iotProject2\AsyncAPI\analytics>npm start

> analytics-service@1.0.0 start
> node src/api/index.js

 SUB  Subscribed to sensor_dummy/values
 PUB  Will eventually publish to analytics/values
 SUB  Subscribed to eKuiper/anomalies
AnalyticsService 1.0.0 is ready!

   MQTT adapter is connected!
← sensor_dummy/values was received:
{
  created_at: '10/10/2021',
  entry_id: '48',
  TEMPERATURE: '27.0625',
  TURBIDITY: '100',
  'DISOLVED OXYGEN': '0',
  pH: '6.05951',
  AMMONIA: '2E-05',
  NITRATE: '173',
  Population: '50',
  Length: '32.01',
  Weight: '275.9'
}
```