



Универзитет „Св. Кирил и Методиј“ во Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Проект по предметот веб програмирање

IOT-Hub

Изработил: Кристијан Постолов 151067

Contents

Опис на проектот.....	3
Користени технологии	3
Стартување на проектот	4
Функционалности.....	4
Автентикација.....	5
Преглед на уреди	5
Детален приказ на уред.....	7
Графици.....	10
Користење на симулатор.....	11
Конфигурациска порака.....	11
Порака за статус.....	12

Опис на проектот

IOT-Hub е веб апликација на која корисниците ќе можат да прегледуваат и праќаат податоци на нивните *iot*(internet of things) уреди. За комуникација со уредите, апликацијата користи MQTT протокол за праќање на пораки. Притоа, за да може уредот да комуницира со системот, првин треба да се програмира според интерфејсот дефиниран од IOT-Hub за да праќа и прима пораки до MQTT серверот.

Корисниците би можеле да ги извршуваат следниве операции:

- Регистрација и најава на апликацијата
- Креирање на уред во системот – внесува име и опис на уредот, а апликацијата генерира идентификатор со чија помош уредот ќе комуницира со системот.
- Преглед и пребарување на уредите кои ги креирал корисникот – корисникот ќе може да ги погледне деталите за уредите како што се име, опис, идентификатор, статус (*inactive*, *paired*, *active*), статус на параметрите што ги има уредот (сензори и актуатори)
- Менување на вредност на параметри – корисникот преку апликацијата би можел да смени вредност на некој од актуаторите (пр. Вклучување на клима преку уредот).
- Преглед на параметрите во форма на график

Уредите би можеле да комуницираат со системот на следниот начин:

- Config порака – употребувајќи го идентификаторот кој го добил корисникот при креирање на уредот, праќа порака за конфигурирање на уредот. Во оваа порака уредот ги праќа своите параметри (име на параметар, тип на податок, дали може да се контролира преку апликацијата). Оваа порака би го сменила статусот на уредот од *inactive* во *paired*. Пораката би можела да се прати повторно за конфигурирање на нови параметри.
- State порака – порака со која уредот праќа нов статус за еден или повеќе параметри (читање на сензор, менување на состојба на актуатор).
- StatePush порака – оваа порака ја праќа системот до уредот, и означува дека корисникот ја променил состојбата на некој од параметрите. (пр. Вклучување на светлина).

Користени технологии

При имплементација на апликацијата опфатени се следниве технологии:

1. Spring boot – за backend модулот од проектот со кој комуницираат корисниците преку frontend апликацијата и уредите преку mqtt протокол
2. Angular – за frontend порталот кој го користат корисниците за прегледување на податоци од своите уреди и испраќање на команди до истите
3. Mosquitto broker – сервер преку кои би комуницирале backend-от и уредите
4. Eclipse paho – библиотека за mqtt комуникација
5. Spring security – автентикација базирана на JWT (json web token) за автентификација/авторизирање на корисниците преку HTTP
6. H2 и Postgre – бази на податоци, можност за користење на апликацијата без конфигурирање на посебна база со помош на h2
7. Chart.js – библиотека за приказ на графови

Стартување на проектот

Проектот може да се најде на <https://github.com/KristijanPostolov/iot-hub> . Во овој проект се наоѓаат backend, frontend и device-simulator(за тестирање на системот без вистински физички iot уреди) модули. Пред да се стартуваат апликациите, треба на локалната машина да имаме инсталирано и стартувано Mosquitto broker. Овој процес е едноставен и е опишан на <https://www.roytuts.com/how-to-install-mosquitto-mqtt-in-windows/> или <https://www.vultr.com/docs/how-to-install-mosquitto-mqtt-broker-server-on-ubuntu-16-04> (многу лесно може да се најде истото и за други оперативни системи). Овој сервер се стартува на порта 1883, но доколку се стартува на друга порта, тоа треба да се смени и во application.properties (mqtt.connectionUrl) на backend модулот и на симулаторот.

Возможно е да се стартува проектот и без mosquito, но со ограничени функционалности (автентикација, креирање и преглед на уреди) со тоа што сервисот воопшто не би комуницирал со уредите(симулаторите). За да се постигне ова, во application.properties треба да се смени полето mqtt.enabled во false.

Проектот е поставен да работи со h2 база (in memory), затоа не е потребно да се креира нова база, но тоа значи дека при рестартирање на backend модулот, сите податоци се губат. За да работи со PostgreSQL база, доволно е да се направи промена во application.properties со тоа што ќе се избришат конекциските параметри за h2, а ќе се од-коментираат параметрите за postgres и соодветно ќе се внесат конекцискиот стринг и credentials.

За frontend модулот, ќе треба да се отвори терминал во модулот и да се внесе npm install, за да се симнат бараните библиотеки.

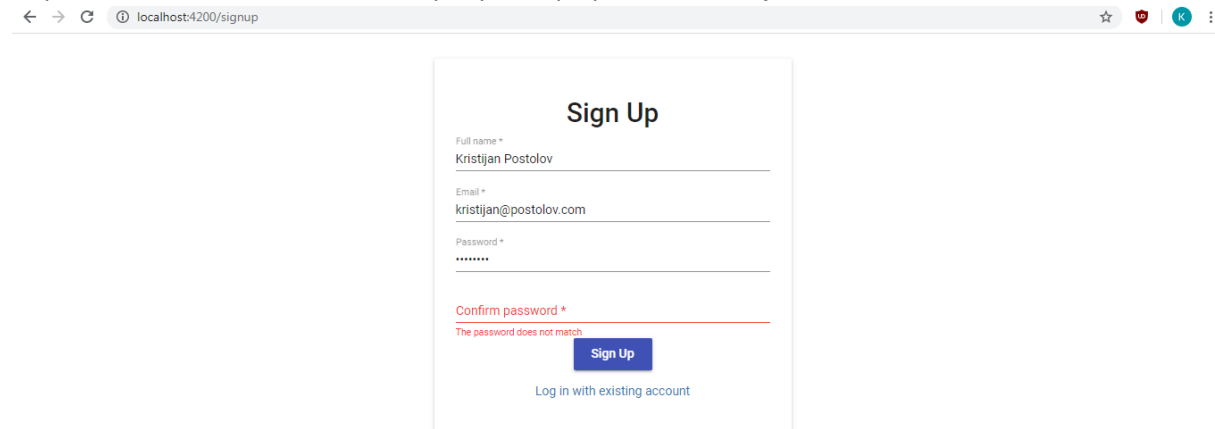
За стартување на frontend и backend апликациите, веќе има поставени shared run configurations за intellij и единствено треба да се клинке run или debug. Истото важи и за симулаторот.

Функционалности

Автентикација

Апликацијата со помош на spring security има имплементирано jwt based authentication.

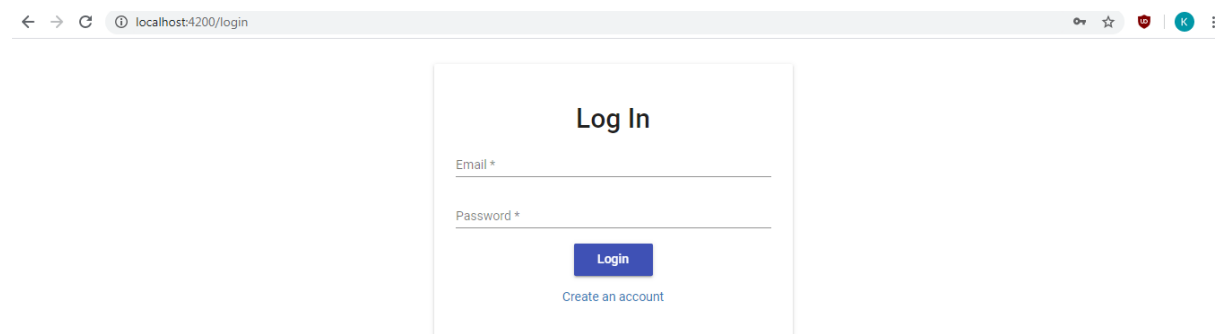
Корисниците имаат можност да креираат профил и да се најават.



The screenshot shows a web browser window with the address bar displaying 'localhost:4200/signup'. The page title is 'Sign Up'. The form contains the following fields and elements:

- Full name ***: Input field with the value 'Kristijan Postolov'.
- Email ***: Input field with the value 'kristijan@postolov.com'.
- Password ***: Input field with masked characters '*****'.
- Confirm password ***: Input field with a red error message below it: 'The password does not match'.
- Sign Up**: A blue button.
- Log in with existing account**: A link below the button.

По креирање на профил, веднаш се редиректираат на главната страна. Доколку не е автентициран корисникот, првин ќе се најде на страната за најава:

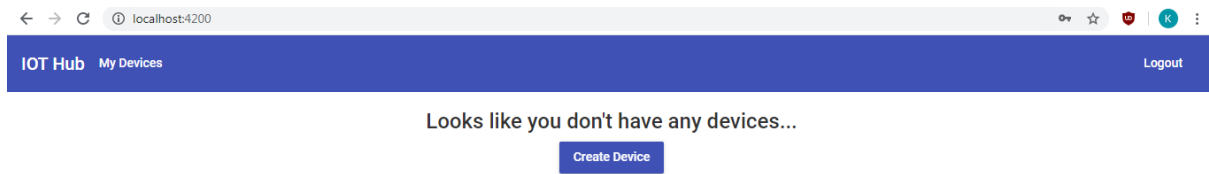


The screenshot shows a web browser window with the address bar displaying 'localhost:4200/login'. The page title is 'Log In'. The form contains the following fields and elements:

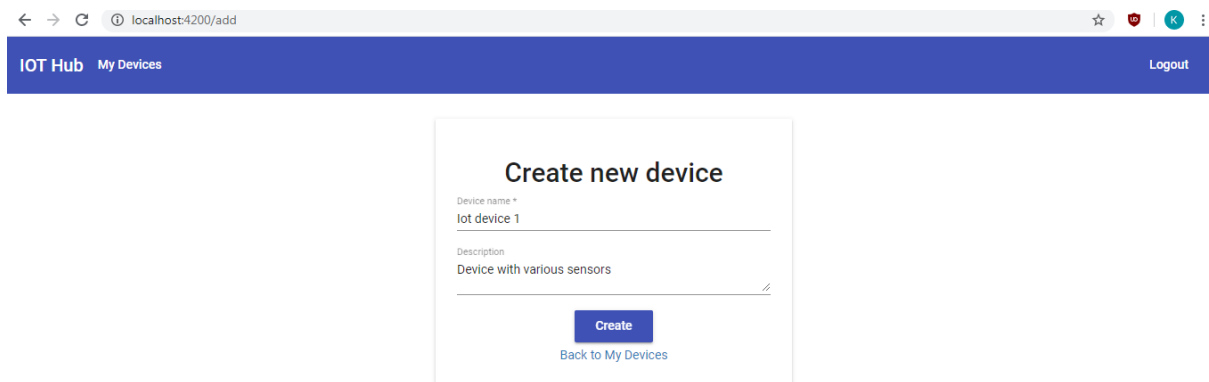
- Email ***: Input field.
- Password ***: Input field.
- Login**: A blue button.
- Create an account**: A link below the button.

Преглед на уреди

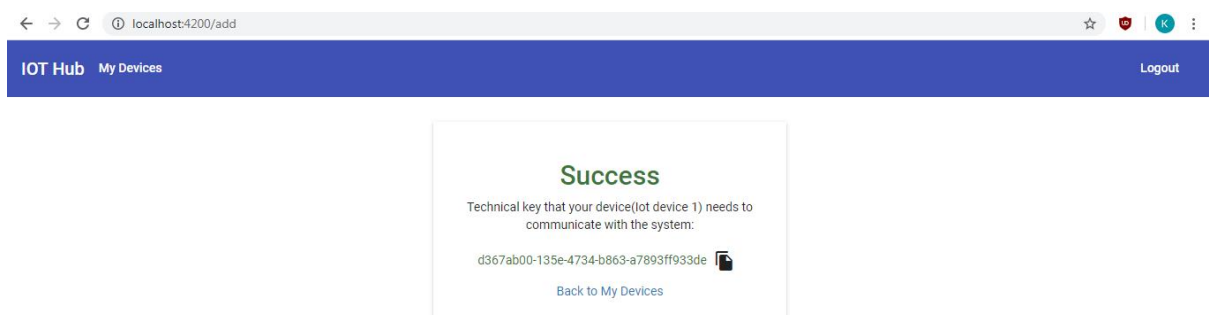
Кога за прв пат корисникот се најде на главната страна, ќе добие порака дека нема уреди, со линк до формата за креирање на нов уред. Со клик на logout копчето, корисникот се враќа на страната за најава (при што се прави revoke на jwt токенот).



Со клик на копчето Create device, на корисникот му се прикажува форма за креирање на уред:

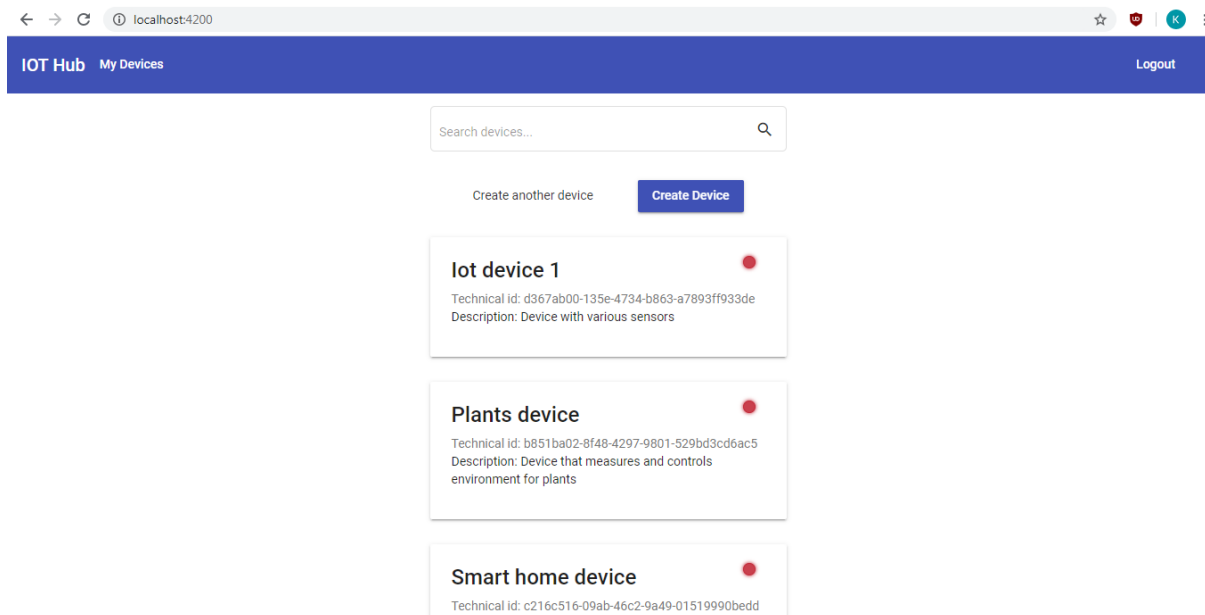


При креирање на уред, корисникот внесува податоци за име на уредот (по кој ќе може потоа да го пребарува) и опис.



Системот прикажува порака дека уредот е креиран и генерира идентификатор кој корисникот ќе го внесе во уредот за тој да може да комуницира со сервисот. При клик на иконата што се наоѓа десно од идентификаторот, се копира идентификаторот (copy to clipboard функционалност) и се прикажува соодветна порака.

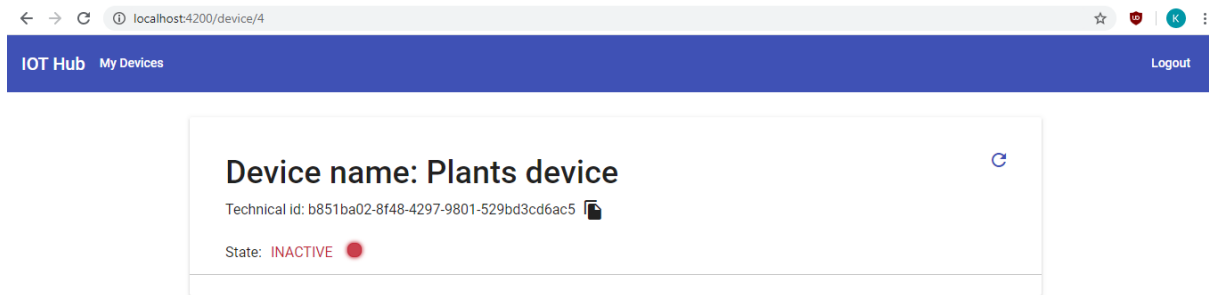
Откако корисникот ќе внесе неколку уреди, сега на почетната страна ќе може да ги види истите:



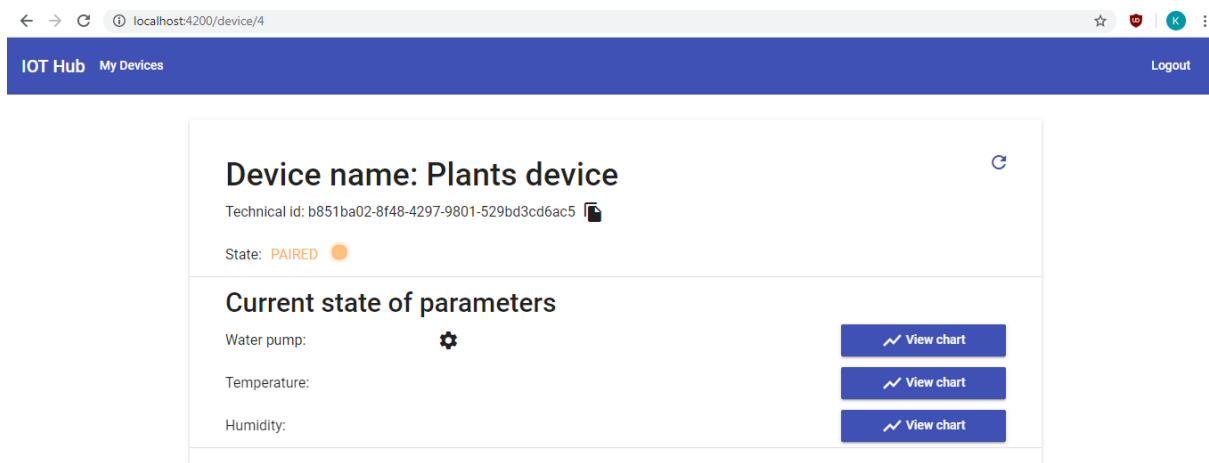
За секој од уредите овде се прикажува име, опис, идентификатор и статус индикатор (црвена точка) кој сега е inactive, бидејќи уредите се само креирани во системот, а сеуште не комуницираат. Исто така, има опција за пребарување по име.

Детален приказ на уред

Со клик на некој од уредите, отвораме страна со детален приказ на уредот.



Бидејќи уредот уште не е активен, ги имаме истите податоци од претходно, но доколку сега системот од страна на уредот добие порака за конфигурирање, неговиот статус ќе премине во paired и веќе ќе може да ги гледаме параметрите на уредот.



Можеме да забележиме дека уредот има три параметри (temperature, humidity, water pump) кои засега се уште немаат статус, туку само знаеме дека постојат. Може да забележиме дека за water pump имаме опција да ја смениме неговата вредност бидејќи е конфигуриран како актуатор.

Доколку сервисот добие порака за статус на уредот (мерење на сензори), статусот му се менува во active и сега може да ги видиме вредностите на параметрите:

← → ↻ localhost:4200/device/4 ☆ 🔒 K ⋮

IOT Hub My Devices Logout

Device name: Plants device

Technical id: b851ba02-8f48-4297-9801-529bd3cd6ac5

State: **ACTIVE**

Current state of parameters

Water pump:	false	⚙️	View chart
Temperature:	24		View chart
Humidity:	84.2		View chart

Со секоја наредна порака за статус, ќе се менуваат вредностите на параметрите соодветно. Исто така, можно е да се реконфигурира уредот, со добивање на друга порака за конфигурирање:

← → ↻ localhost:4200/device/4 ☆ 🔒 K ⋮

IOT Hub My Devices Logout

Device name: Plants device

Technical id: b851ba02-8f48-4297-9801-529bd3cd6ac5

State: **ACTIVE**

Current state of parameters

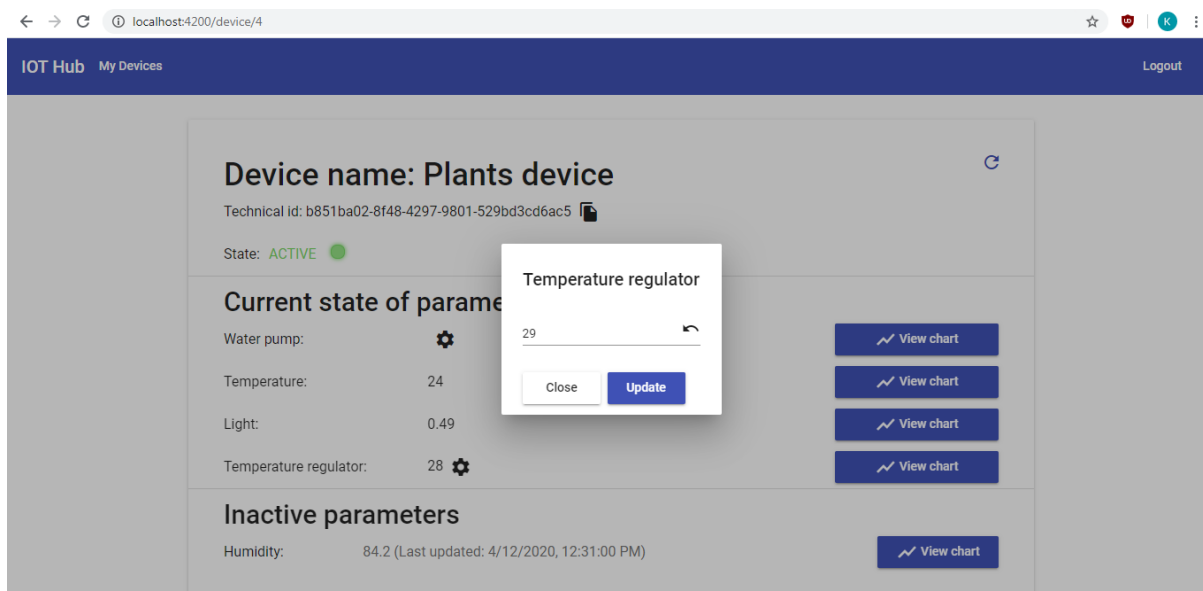
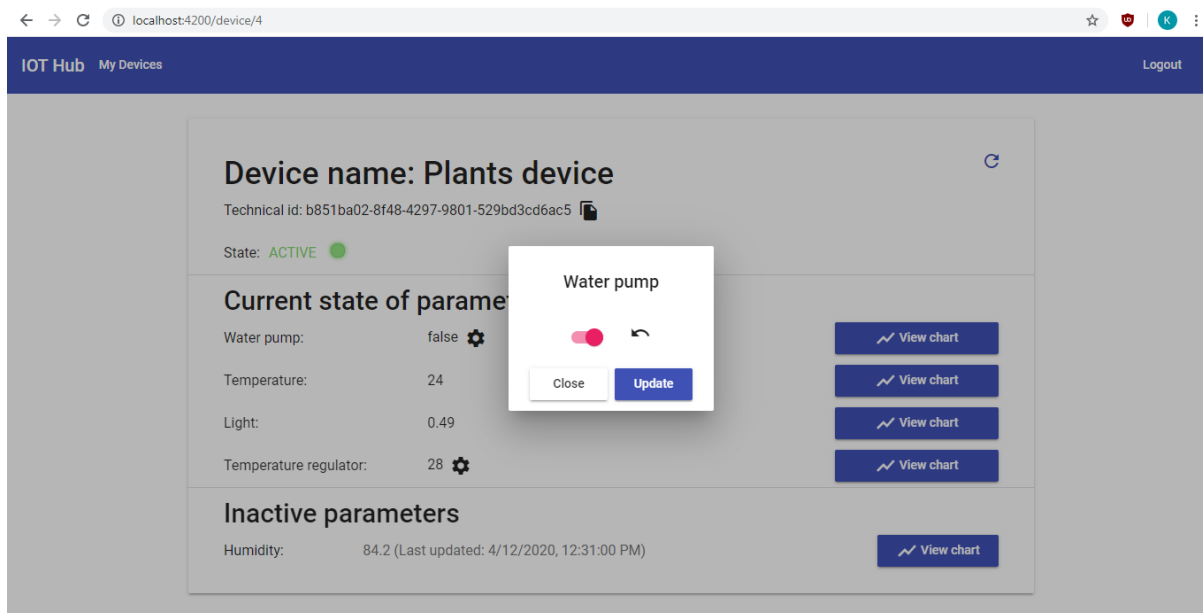
Water pump:	false	⚙️	View chart
Temperature:	24		View chart
Light:	0.49		View chart
Temperature regulator:	28	⚙️	View chart

Inactive parameters

Humidity:	84.2 (Last updated: 4/12/2020, 12:31:00 PM)	View chart
-----------	---	----------------------------

Можеме да забележиме дека сега се додадени нови параметри, а и дека еден од параметрите веќе не е активен.

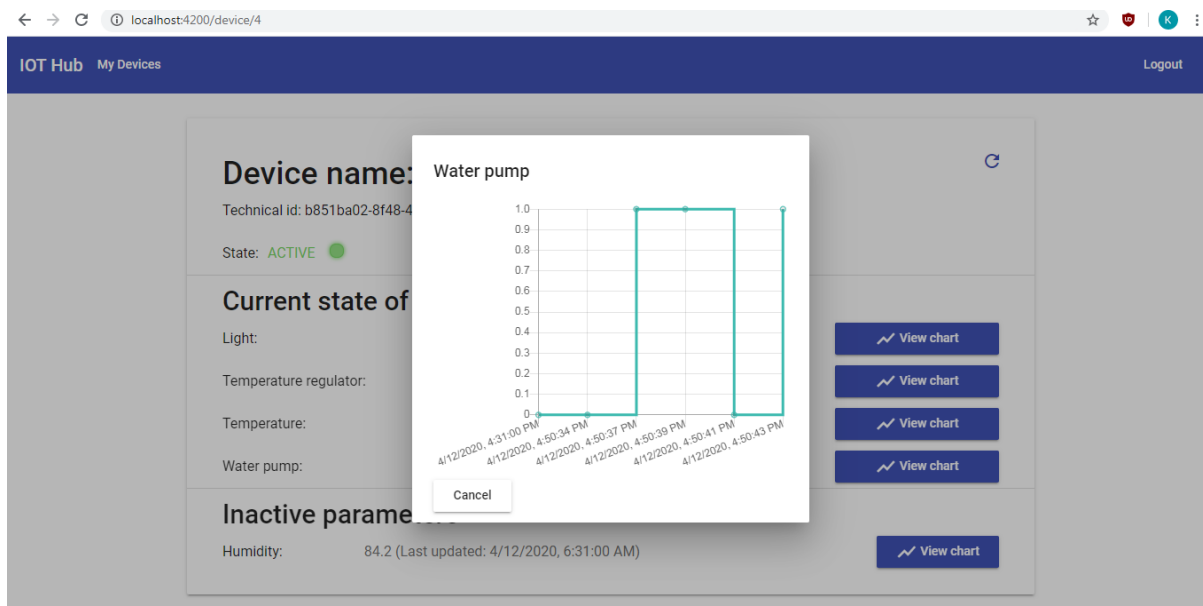
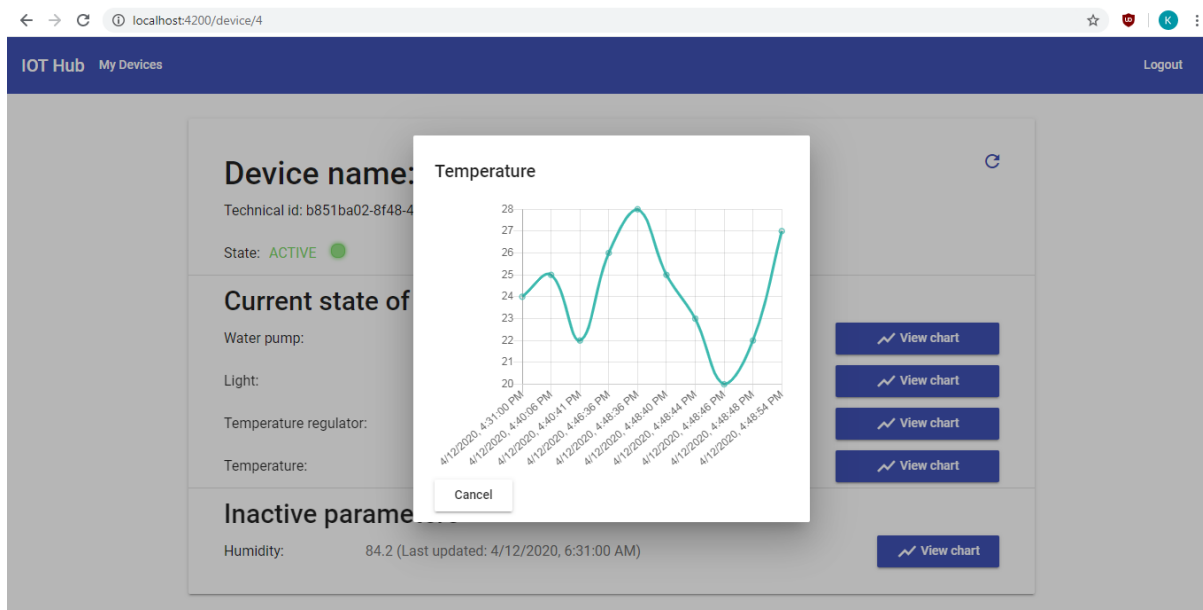
Како што споменавме претходно, за актуаторите имаме опција за менување на вредноста преку апликацијата. Со клик на иконата десно од вредноста на параметарот, добиваме форма за менување на вредноста. Оваа форма е различна за различни типови на параметри (BOOLEAN, INTEGER, REAL).



Со клик на update копчето, сервисот испраќа push state порака до соодветниот уред со помош на mqtt протоколот за менување на статусот на параметарот.

Графици

Со клик на View Charts копчето на некои од параметрите, ќе ни се отвори дијалог со график за вредностите на параметарот во однос на времето.



Користење на симулатор

Како другите компоненти, симулаторот има своја run configuration за лесно стартување. Оваа компонента ни помага да го тестираме системот без вистински уред.

Кога ја стартуваме оваа компонента, првин во конзола добиваме порака за внесување на идентификаторот на уредот (што корисникот го добива при креирање на уредот преку апликацијата).

Потоа, добиваме порака со упатство како да пратиме некоја порака.

Конфигурациска порака

Доколку сакаме да пратиме конфигурациска порака, шаблонот е:

```
config;<name>,<type>,<actuator>
```

со што, може да внесеме повеќе од еден параметар со одделување со ; карактерот. <type> делот може да има вредности BOOLEAN, INTEGER или REAL. <actuator> може да има вредност true или false.

Пример: config;temperature,REAL,false;water_pump,BOOLEAN,true

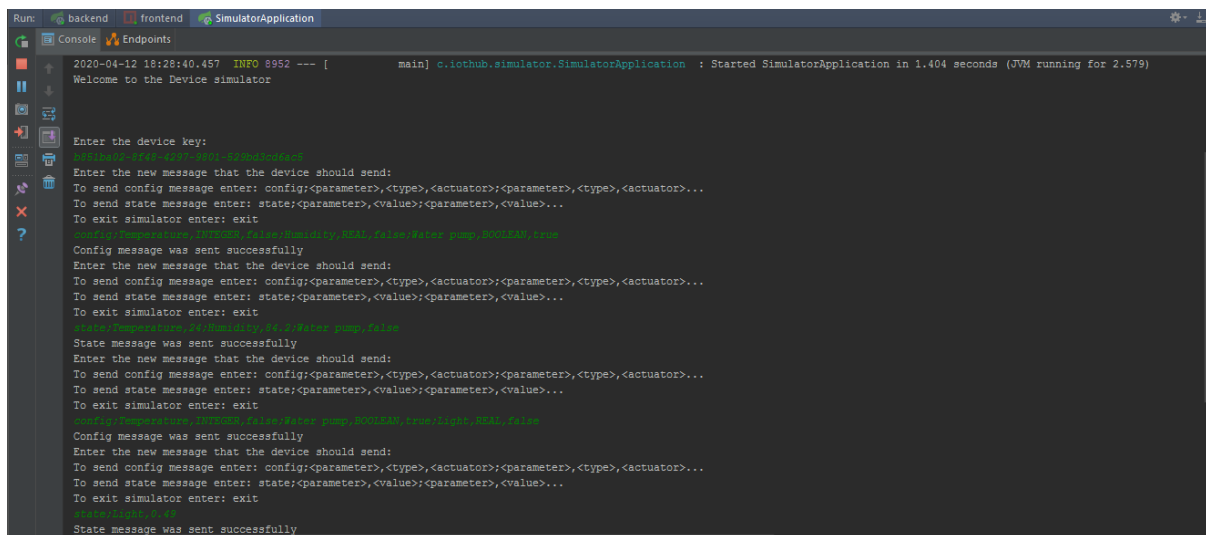
Порака за статус

Шаблон за порака за статус е:

state;<name>,<value>

Овде исто така може да внесуваме повеќе параметри, со тоа што на местото на <name> може да стои само некоја од претходно конфигурираните имиња. Кај INTEGER и REAL параметрите на местото на <value> треба да внесеме цел број или реален број, додека за BOOLEAN, само true и false се валидни вредности.

Пример: state;temperature,15.5;water_pump,true



```
Run: backend frontend SimulatorApplication
2020-04-12 18:28:40.457 INFO 8952 --- [main] c.iothub.simulator.SimulatorApplication : Started SimulatorApplication in 1.404 seconds (JVM running for 2.579)
Welcome to the Device simulator

Enter the device key:
Enter the new message that the device should send:
To send config message enter: config;<parameter>,<type>,<actuator>,<parameter>,<type>,<actuator>...
To send state message enter: state;<parameter>,<value>;<parameter>,<value>...
To exit simulator enter: exit
config;temperature,REAL,false;water_pump,BOOLEAN,true
Config message was sent successfully
Enter the new message that the device should send:
To send config message enter: config;<parameter>,<type>,<actuator>,<parameter>,<type>,<actuator>...
To send state message enter: state;<parameter>,<value>;<parameter>,<value>...
To exit simulator enter: exit
state;temperature,15.5;water_pump,true
State message was sent successfully
Enter the new message that the device should send:
To send config message enter: config;<parameter>,<type>,<actuator>,<parameter>,<type>,<actuator>...
To send state message enter: state;<parameter>,<value>;<parameter>,<value>...
To exit simulator enter: exit
state;temperature,15.5;water_pump,true
Config message was sent successfully
Enter the new message that the device should send:
To send config message enter: config;<parameter>,<type>,<actuator>,<parameter>,<type>,<actuator>...
To send state message enter: state;<parameter>,<value>;<parameter>,<value>...
To exit simulator enter: exit
state;temperature,15.5;water_pump,true
State message was sent successfully
```