HOME IOT MONITORING AND MANAGEMENT SYSTEM

Dimitar Nikolov, Nikolay Pavlov, Asen Rahnev

Abstract. This article describes platform for monitoring and managing sensors and devices for home. Protocol for data transfer from and to devices was created to support managing and monitoring. Web portal was created where users can see the current information for all their devices and create queries and analysis over past periods. Information can be exported in different formats if additional analysis are required.

Keywords: Real Time monitoring, IoT

1. Introduction

Nowadays there is enough cheap hardware that supports IoT however receiving and storing information from devices is often left to the user. Hardware that comes with integrated managing system is expensive and is not used on large scale. Therefore platforms that supports receiving, storage and managing of IoT devices are rising. However those platforms are mostly targeted at large customers, or even created for single business deal. The protocol and web portal that this document describes are targeted at individuals who want to implement home monitoring or automation, without charging them high prices that could ruin the effect of automation.

2. Functionality

All functionalities are available only for registered users. This step is required to avoid data mashup for different users of the platform. Registered users can register

devices, sensors, monitor the received data real-time or look at data received for past periods.

User Registration

In order to use the platform users must register first. This is accomplished with simple user registration form in the portal requiring only email and password. Registration is required in order to avoid data mashups and to protect the portal from basic attacks. However registration is the only step that needs to be made within the web portal, if the user wants he/she can later on use the provided REST API to register, retrieve and delete data.

Device Registration

Automatic registration

After the user is registered successfully he/she can start sending data to the portal without doing any other preparation and within. The provided api for data storing has all required fields for saving information regarding the device, sensor and sent data. The bare minimum information is packed in an object with properties for sessionid and array of sensor data (sensor identifier and value). Device identifier is optional field and if it is provided and not existing in the portal new device with such identifier is created.

Manual registration

Using the provided web site the user can register devices and sensors before sending any data from them. Devices and sensors are matched based on device identifier and sensor identifier respectively.

Real time monitoring

Real time data

The web site provides clean and simple UI to monitor received data in close to real-time. This functionality is accomplished with notifications sent through web sockets.

Data processing

There is built in functionality to further instantaneous data processing. However, this is accomplished by asynchronous analysis of the received data. In IoT world long lasting connections could degrade battery life significantly. That is why processing of the data is made on the server after response is sent to the device.

Alarms

Alarms is an item in the API that defines the processing of received data. All received samples are matched against user defined conditions (greater than, less than, equals, etc). When the data sample does match the condition defined in the

alarm processing stops. If alarm condition is met (let's say value equals 10) processing continues. There are two scenarios. First scenario is to send notification (currently only notifications are sent only through email). The second scenario is to send action definition to the device.

Actions

Action is user defined data that will be sent to device when requested. The user could define action which is sent to the device as binary or string. Devices handle the returned response as desired by user. Actions allow autonomous control of IoT devices. This functionality saves memory (often very limited) and provide device re-usage without reprogramming. Users can create and send actions from the web portal or through the API.

Reports

The web portal uses the API to provide simple reports with graphical presentation of data changes through time. Reports could be made for real-time data or cold data. The API beneath requires from date, to date and sensor.

- Raw data report
  Raw data report is designed to show all data as received from sensor. The report displays all received data table and provides .xlsx download.
- Reports with Aggregations
  Report with aggregations provides functionality to have sum, average, minimum, maximum of the received data.
- Comparison reports
  Report displays data for two or more sensors for the given period providing the user comparison of received data.

3. Technical description

The system leverages the ease and functionalities provided by .Net Core [1]. .NET Core [2] provides latest technologies and simple yet powerful API.

API

The API of the system is implemented in with ASP.Net Core. For data storage the system uses MSSQL server [3] and/or Firebird [1]. More detailed functional and technical description will be provided for the API in other articles. The API is separated into several smaller interfaces.

IApi

This interface combines all provided interfaces. It also defines the basic actions provided login, receiveData and GetSensorData.

```
public interface IApi : ISensors, IAlarms, IActions, IDevices
```

```
{
    LoginResponse Login(LoginRequest request);
    ReceiveDataResponse ReceiveData(ReceiveDataRequest request);
    GetSensorDataResponse GetSensorData(GetSensorDataRequest request);
}
```

ISensors

This interface provides all functionalities concerning sensors

```
public interface ISensors
{
    GetUserSensorsResponse GetUserSensors(string sessionId);
    Sensor SaveSensor(SaveSensorRequest request);
    DeleteItemResponse DeleteSensor(DeleteRequest request);
}
```

IAlarms

This interface provides all functionalities for managing alarms.

```
public interface IAlarms
{
    GetAlarmsResponse GetAlarms(string sessionId);
    Alarm SaveAlarm(SaveAlarmRequest request);
    DeleteItemResponse DeleteAlarm(DeleteRequest request);
    GetAlarmMatchersResponse GetAlarmMatchers(string sessionId);
    GetAlarmTypesResponse GetAlarmTypes(string sessionId);
}
```

IActions

This interface combines functionalities for sending actions/commands to devices.

```
public interface IActions
{
    GetActionsResponse GetActions(string sessionId);
    Domain.IoTApi.Actions.Action SaveAction(SaveActionRequest request);
    DeleteItemResponse DeleteAction(DeleteRequest request);
    ActionExecution ExecuteAction(ExecuteActionRequest request);
    GetDeviceActionsResponse    GetDeviceActions(GetDeviceActionsRequest
request);
    GetActionExecutionsResponse
GetActionExecutions(GetActionExecutionsRequest request);
```

```
}
```

IDevices

This interface provides functionalities for managing devices.

```
public interface IDevices
{
    GetDevicesResponse GetDevices(string sessionId);
    Device SaveDevice(SaveDeviceRequest request);
    DeleteItemResponse DeleteDevice(DeleteRequest request);
    object Ping(string deviceIdentifier, string sessionId);
}
```

Communication Protocol

The provided API is REST json. Http and Https are used for communication. While requiring power hungry chips Http could also be accessed from Gateways like these provided by LoraWAN [7] and Sigfox [8]. In general, this approach does not limit the hardware implementations, but provides unification and more freedom in upper application layers either desktop, web or mobile.

Web portal

Web portal is implemented as Single Page Application using only the provided API. AngularJs [4] was used to create the portal, however upgrade to Angular 5 is foreseen. Other technologies used: Gulp [6], Bootstrap [5].

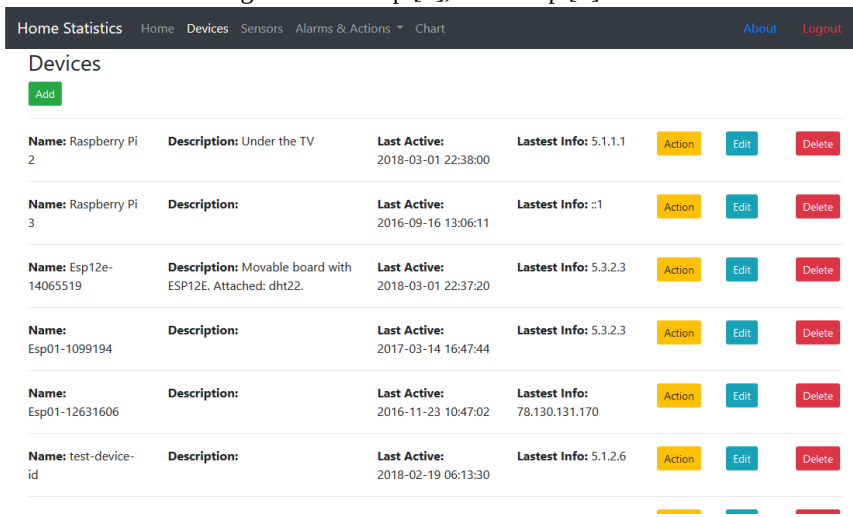

Figure 1. Devices screen

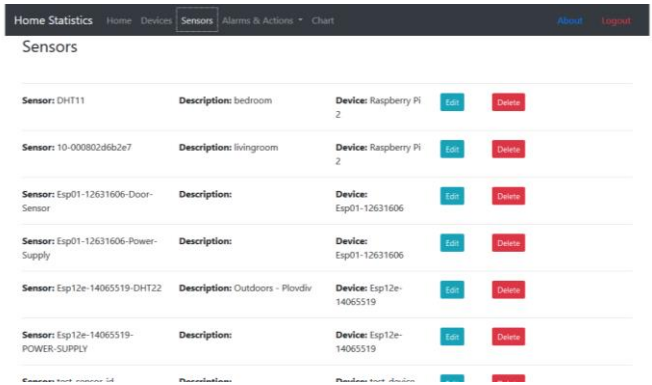

Figure 2. Sensors screen

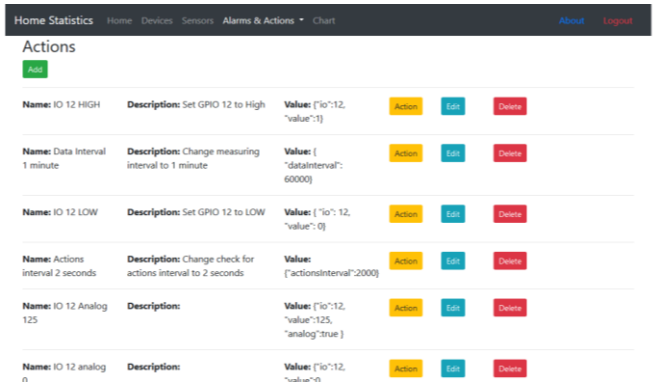

Figure 3. Actions screen

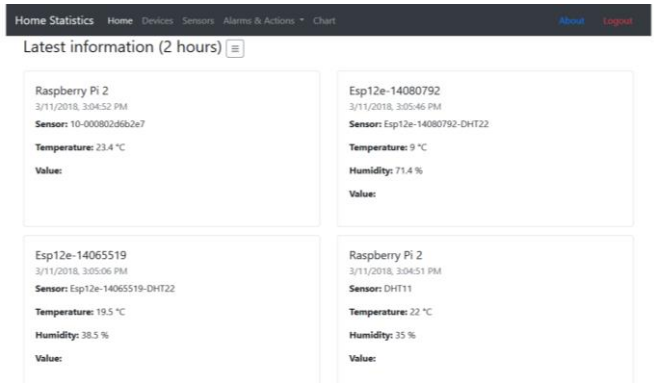

Figure 4. Home screen

4. Conclusion

In this article we have description of idea and solution for common problems in home automation like data storage, retrieval and automatic processing. Being multi-tenant by design the solution could be used by thousands of users in the cloud or as a stand-alone server in a household. This article also gives good example for usage of powerful technologies and their mainstream and eccentric combinations.

Acknowledgements

This work is partially supported by the FP17-FMI-008 project of the Scientific Fund of the University of Plovdiv "Paisii Hilendarski", Bulgaria.

References

[1] Introduction to ASP.NET Core 2.0, https://docs.microsoft.com/en-us/aspnet/core/, retrieved on January 2018
[2] SQL Server Documentation, https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation, retrieved on January 2018
[3] Firebird Documentation, http://www.firebirdsql.org/en/documentation/, retrieved on January 2018
[4] AngularJS Documentation, https://docs.angularjs.org/guide, retrieved on January 2018
[5] Bootstrap Documentation, http://getbootstrap.com/docs/4.0/getting-started/introduction/, retrieved on January 2018
[6] Gulp API, https://github.com/gulpjs/gulp/blob/v3.9.1/docs/API.md, retrieved on January 2018
[7] LoraWAN, https://lora-alliance.org
[8] Sigfox Services, https://www.sigfox.com/en

Faculty of Mathematics and Informatics
Plovdiv University
236 Bulgaria Blvd,
Plovdiv 4003, Bulgaria
E-mails: d.nikolov@kodar.net, nikolayp@uni-plovdiv.bg, assen@uni-plovdiv.bg

IoT Система за управление на „умни“ къщи

Димитър Николов, Николай Павлов, Асен Рахнев

Резюме. Работата представя платформа за наблюдение и управление на сензорни устройства за дома. Създаден е протокол за обмен на данни и управление на устройствата. Изграден е уеб портал, чрез който потребителите могат да получат текуща информация за състоянието на наблюдавания от тях обект и да изготвят справки за изминали периоди. Информацията може да се експортира в различни формати за допълнителни анализи.

Ключови думи: IoT, Мониторинг в реално време