



# A Web Platform for Globally Interconnected 6LoWPANs

David Janowsky

February 2019



## Introduction

- ▶ Why 6LoWPAN?
- ▶ Why CoAP?
- ▶ Web Application



## Why 6LoWPAN?

- ▶ IPv4 address space is running out
- ▶ More and more smart objects, embedded systems, wireless devices
  - ▶ Need to run many on low power and with minimal resources
- ▶ Want to be able to use IP with these devices
  - ▶ Can integrate into IoT



## Why CoAP?

- ▶ Use UDP over TCP
  - ▶ Want shorter delay time
- ▶ Specially designed for constrained networks
- ▶ Can easily interface with HTTP

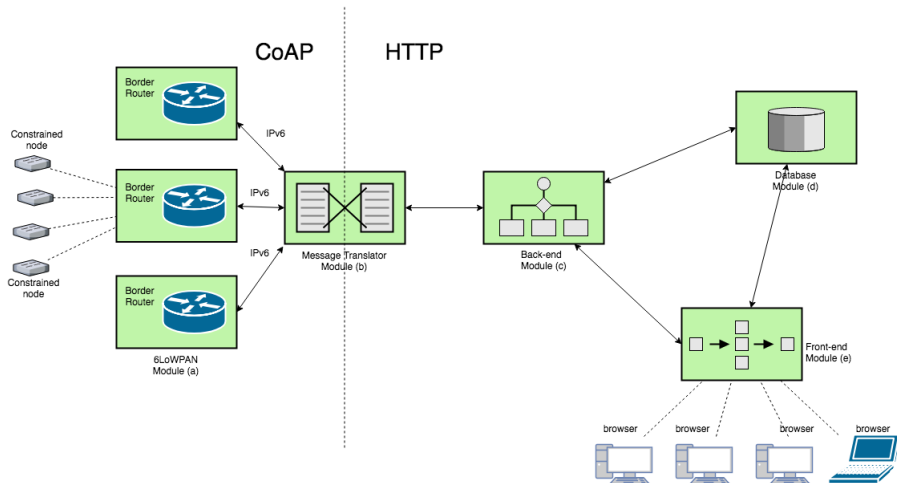


## Web Application

- ▶ Interfaces with 6LoWPAN platform
- ▶ Connects clients over HTTP
- ▶ Connects 6LoWPAN/IoT devices over CoAP
- ▶ Modules
  - ▶ 6LoWPAN devices
  - ▶ Translating module
  - ▶ Back-end
  - ▶ Database
  - ▶ Front-end



## Web Application





## Design and Implementation Overview

- ▶ RESTful architecture with standard HTTP methods
- ▶ Front-end technologies
  - ▶ HTML, CSS, JavaScript/jQuery, Bootstrap, Java Server Page (JSP)
- ▶ Back-end technologies
  - ▶ Java Spring MVC, logical layer
- ▶ Database technologies
  - ▶ Hibernate, MySQL



## Design

- ▶ Necessary functionality
- ▶ Driven by user experience
- ▶ 8 cases were considered



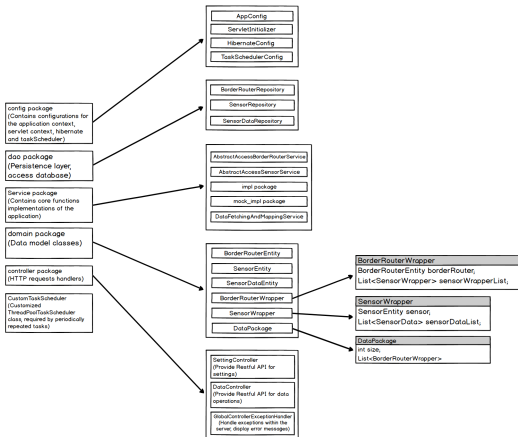


## Use Cases

1. Application must communicate with sensor network over multiple protocols
2. Users can easily access application through a GUI in a browser
3. Users can add border routers in the GUI
4. Server can automatically add sensor nodes on the sensor network
5. Users can set data types that sensors report
6. Users can monitor data from each sensor
7. Users can toggle automatic data fetching and can see data trends over time
8. Users can start new sessions or continue from a previous session

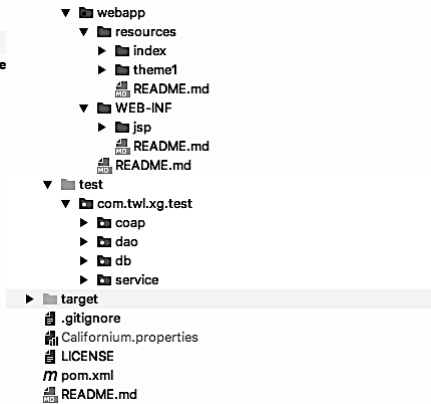
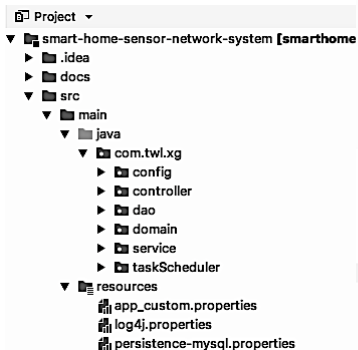


## Back-End Class Structure



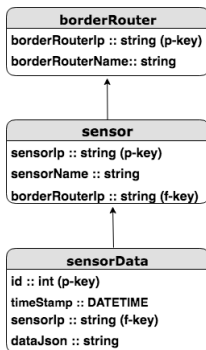


## Implementation





## Database Entity Relation





## Index Page





# Index Page

It is easy to use!

Just type in your border router IPv6 address, our application will find all sensors connected to your border router automatically. All our interfaces are self-configured, just click your sensor and see what's going on in your house!

[GET STARTED](#) [CONTINUE](#)

### House Monitor Sensor Network System

This project is a full-stack project that can communicate with sensor networks using CoAP protocol. It can collect environmental data sets from sensors and store them in database. The back-end part provides a set of RESTful API that are easy to program with. Other developers can use this set of API to their Android, iOS or Web applications. See [README](#) for more information. Also, the application has a front-end part that communicates with the back-end part via RESTful API. The front-end pages allow users to set up the application and monitor data from a visual interface in their browser.

[VIEW SOURCE CODE](#)

### Let's Get In Touch!

This is the Master's project of Xiaoheng Gao and Xiaofei Gao in Tufts University. If you have any question or advice, please feel free to contact us.



[xgao@tufts@gmail.com](mailto:xgao@tufts@gmail.com)



[dfliangf@gmail.com](mailto:dfliangf@gmail.com)



## Data Type Settings

### Set Data Type

Enter data types you want to monitor. The data type you entered must be supported by sensors you are using. For example, if you want to monitor **temperature**, you might need to enter **temp** for it.

If you want to use the default **dataTypeIdList** that can be configured in "app\_custom.properties" file, click **Skip** button. Currently, the default dataTypeIdList is: **[d1, d2, d3, d4]** .



Press **+** to add another data type

Submit

Skip



## Border Router Settings

### Set Border Router

Fill in this form to set up each [**BorderRouterIP**, **BorderRouterName**] pair. Duplicated border router IP will be ignored. Border router name can be duplicated, but it is not recommended.



Press + to add another border router





## Sensor Name Settings

### Set Sensor Name

Fill in this form to set name of each sensor. Sensor name can be duplicated but this is not recommended. The name of each sensor should be meaningful, for example, use **Kitchen** or **Bedroom** for sensors in your kitchen and bedroom respectively. If you **skip** this step, the name of each sensor will be its IP address.

borderlp--1--sensor-0

Sensor Name

borderlp--2--sensor-0

Sensor Name

borderlp--2--sensor-1

Sensor Name

borderlp--2--sensor-2

Sensor Name

Submit

Skip

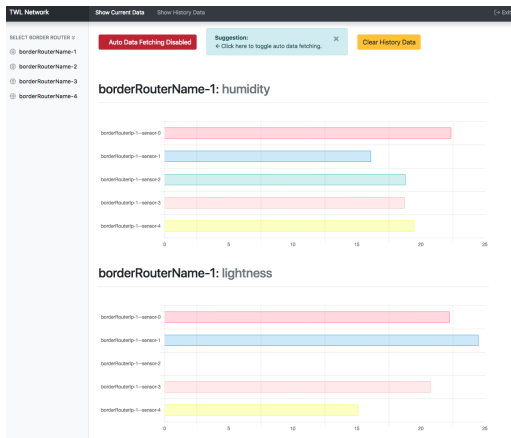


## Auto Data Fetching Settings

The screenshot shows the TWL Network interface. On the left, there is a sidebar with the text "SELECT BORDER ROUTER" and a list of four items: "borderRouterName-1", "borderRouterName-2", "borderRouterName-3", and "borderRouterName-4". The main area has a dark header with "TWL Network" on the left and "Show Current Data" and "Show History Data" in the center. A green button labeled "Auto Data Fetching" is visible. A white dialog box titled "Please Set Period" is open in the center, containing a text input field with the placeholder text "period in millisecond" and a blue button labeled "Enable Auto Data Fetching". To the right of the dialog, a yellow button labeled "Clear History Data" is visible. In the top right corner of the interface, there is a small link that says "(→) Exit".

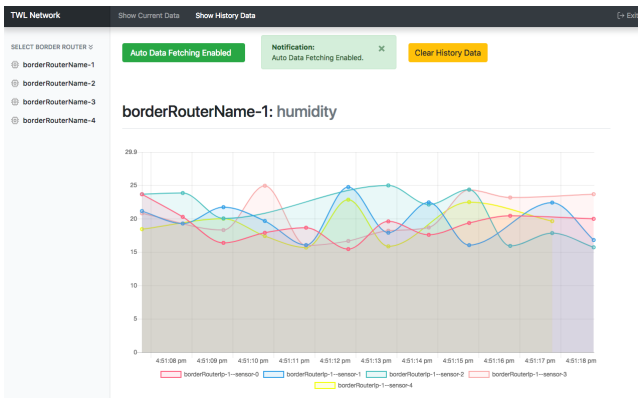


## Results: Current Data



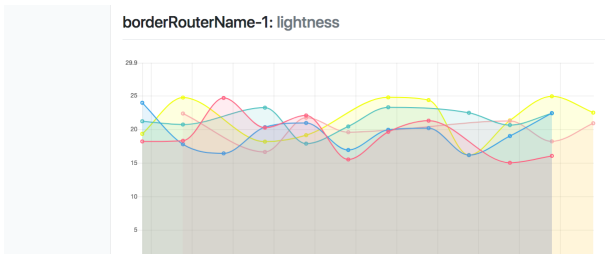


## Results: Historical Data





## Results: Historical Data





## Conclusion

- ▶ Users can set up environment to fit all their specific needs
- ▶ Front-end, back-end and database modules are well functioning
- ▶ Third party translating module being used
- ▶ Connects to 6LoWPAN platform via CoAP



## References