

IoT Modelling Environment

Levindo Gabriel Taschetto Neto
02.11.2017

Advisors: Prof. Dr.-Ing. habil. Bernhard Mitschang
M.Sc. Ana Cristina Franco da Silva
Dipl.-Inf. Pascal Hirmer

Universität Stuttgart

Agenda

1. Aim of the Project
2. Utilized Technologies
3. IoT Modelling Tool
4. Roadmap



1. Aim of the Project

- The project has as goal building a platform, where users can have their own devices with components.
- The users may monitor in real time the values' changes in the sensors attached to their devices.



2. Used Technologies

1. Database
2. Backend
3. Frontend

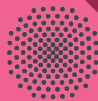


2.1. Database

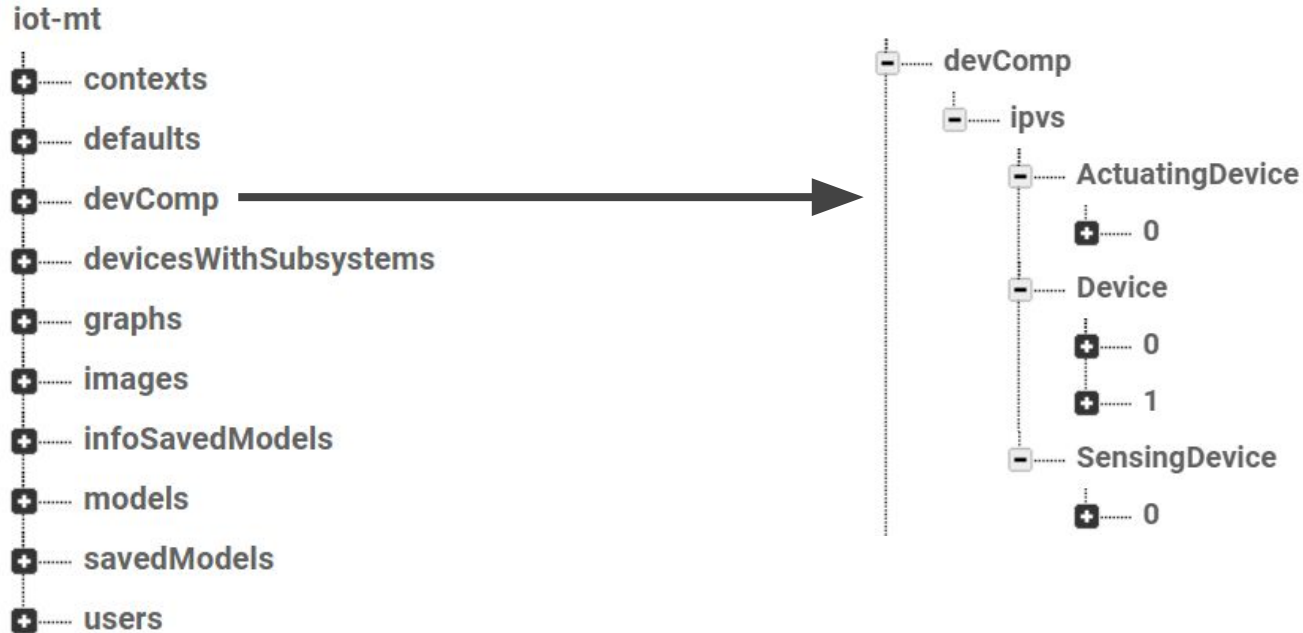
Firebase (Google)

The Firebase Real Time Database is a cloud-hosted NoSQL database that lets data being synchronized and stored between its users in real time.

- Json-based
- Document-store



2.1.1. Firebase Stored Format



2.1.2. Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to the created platform.

Authentication

USERS


SIGN-IN METHOD

TEMPLATES

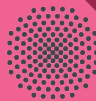
USAGE

Search by email address, phone number or user UID

ADD USER

| Identifier | Providers | Created | Signed In | User UID ↑ |
|----------------------|---|-------------|-------------|------------------------------|
| levindogtn@gmail.com |  | 11 Sep 2017 | 10 Oct 2017 | 4v5v4VAd8cZD3KNCKmaxiGnjTni2 |

```
{  
  "rules": {  
    ".read": "true",  
    ".write": "true"  
  }  
}
```



2.2. Backend

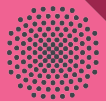
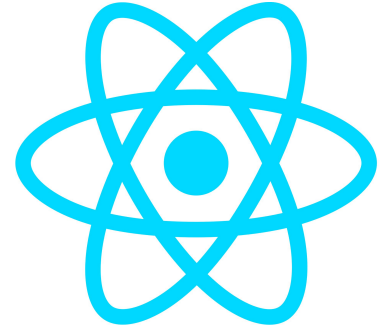
The backend is developed **Javascript**, **Angular JS** (v1.6.6) and **AngularFire** (v2.3.0).



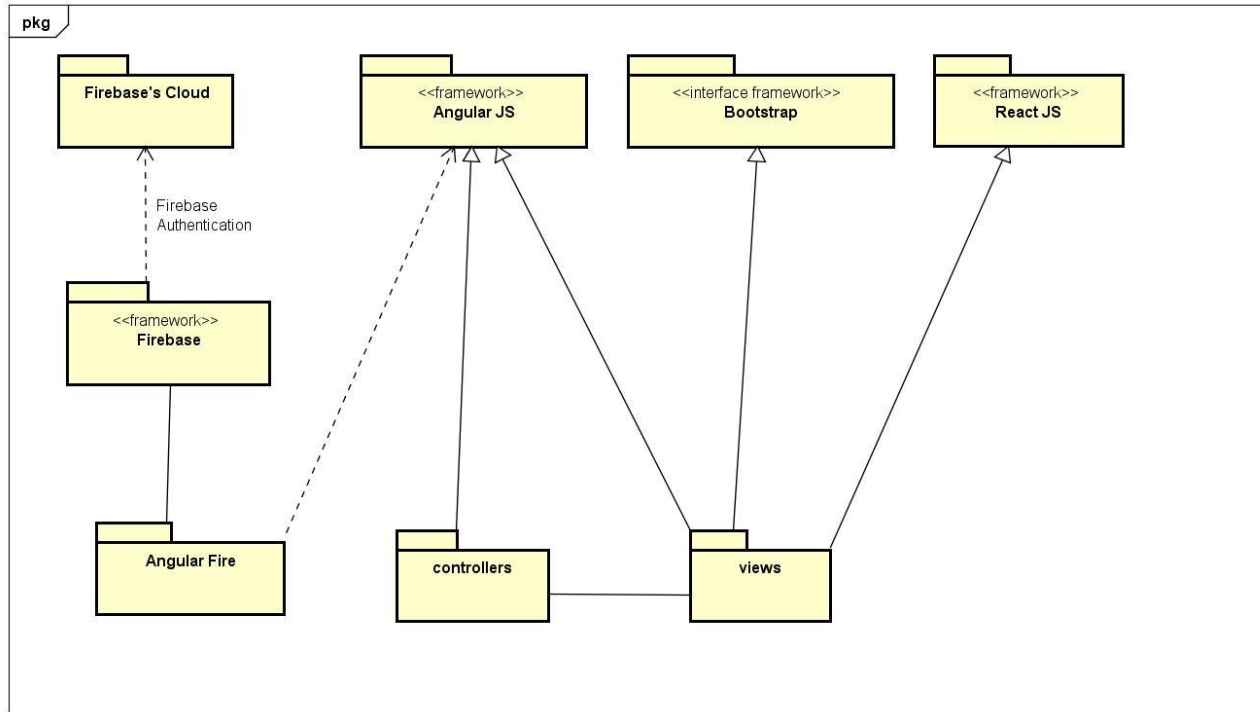
angularFire
by  **Firebase**

2.3. Frontend

The frontend of the application is developed using **HTML** (v5), **CSS** (v3), **Bootstrap** (v3.3.7) and **React JS** (v15.5.4).

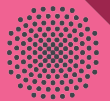


2.4. Relation among the used technologies



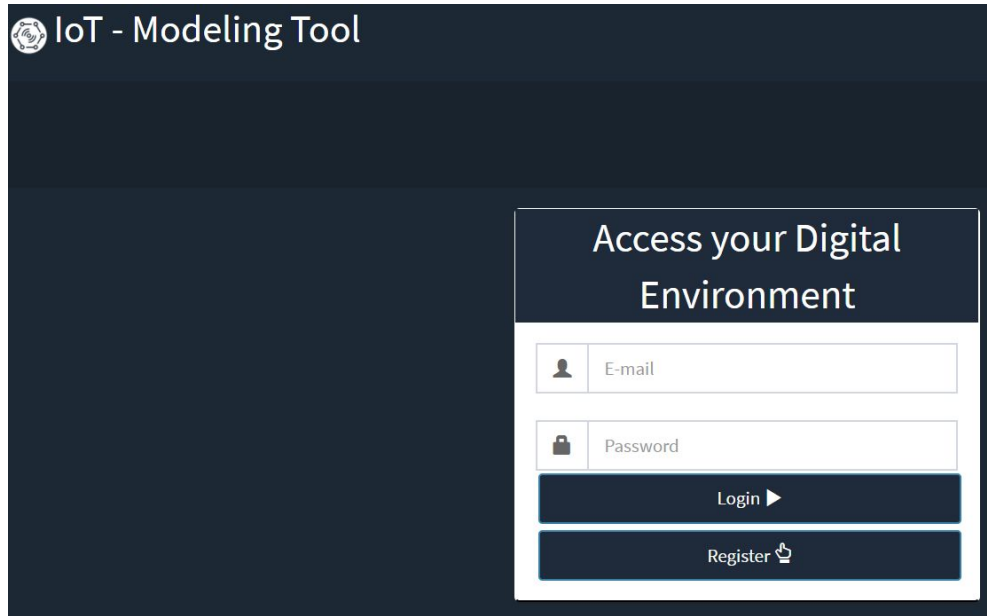
3. IoT Modelling Tool

IoT Modelling Tool is a platform, which allows users to have their own devices and components modeled in order to represent a physical environment.

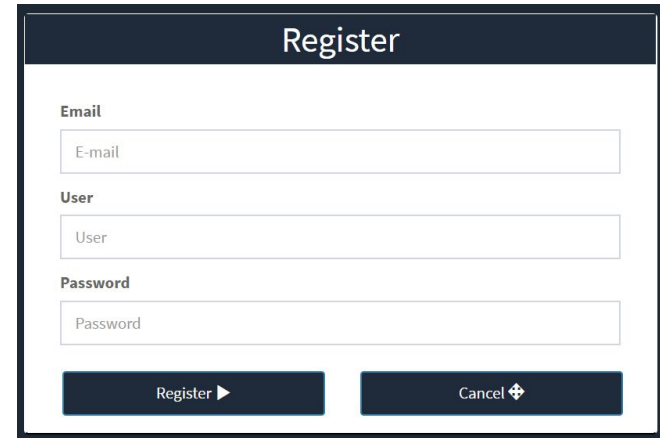


3.1 Registration and Login

The user may register and login onto the platform.



The screenshot shows the main interface of the 'IoT - Modeling Tool'. At the top left is a logo and the text 'IoT - Modeling Tool'. The main area is dark blue. On the right side, there is a white box titled 'Access your Digital Environment'. Inside this box, there are two input fields: one for 'E-mail' with a person icon and one for 'Password' with a lock icon. Below these fields are two buttons: 'Login' with a right arrow and 'Register' with a right arrow and a user icon.



The screenshot shows a 'Register' form. It has a dark blue header with the word 'Register'. Below the header, there are three input fields: 'Email' (with placeholder 'E-mail'), 'User' (with placeholder 'User'), and 'Password' (with placeholder 'Password'). At the bottom, there are two buttons: 'Register' with a right arrow and 'Cancel' with a cross icon.

3.2 Platform's Functionalities

The users can, on the platform:

- Manage their accounts
- Add and manage the IoT Lite Elements (@Context and @Graph)
- Access the IoT Modelling Environment
- Add a device or a component (sensor or actuator)
- Add additional properties in their devices/components
- View detail information about their modeled hardware devices and components
- Search for components and devices



3.3 IoT Modelling Environment

The screenshot displays the 'IoT - Modeling Tool' interface. On the left is a dark blue sidebar with navigation options: 'My Account', 'Add Default @Context', 'Add Specific @Context', 'Add Default @Graph', 'My @Context', 'My @Graph', 'IoT Modelling Environment', 'Add Device', 'My Devices', 'Add Additional Properties', and 'Search'. The main workspace is titled 'IoT Modelling Environment' and features a toolbar with 'SAVE', 'SAVE AS', 'LOAD', 'EXPORT', 'IMPORT', 'CLEAR', and 'HELP' buttons. Below the toolbar is a 'Palette' with categories: 'Devices' (Arduino, RaspberryPi), 'Sensors' (DS18B20), and 'Actuators' (L2G93D). The central canvas shows a hierarchical diagram with a RaspberryPi icon at the top, connected by arrows to three DS18B20 sensor icons below it. A coordinate system '(0,0)' and a '25 cm' scale bar are visible. On the right, a 'Details' panel for the selected DS18B20 sensor shows attributes: '@id: DS18B20-2', 'x: 523', 'y: 322', 'value: 4', 'pinConfiguration' (with a dropdown arrow), and 'isSubSystemOf: RaspberryPi-1'. A link 'See more information' is at the bottom of the details panel.

Information
synchronized with
the database

3.3.1 Navigation Bar

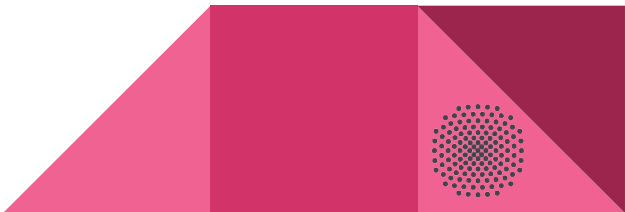
The navigation bar contains synchronized functionalities with the real-time database, as well as importing and exporting models using the file system.

1. Synchronized with Firebase

- 1.1. Save
- 1.2. Save As
- 1.3. Load

2. Independent from Firebase

- 2.1. Export
- 2.2. Import
- 2.3. Clear



3.3.2 Palette Container

The palette contains the currently stored smart devices and components on the database.

It's divided in 3 labels

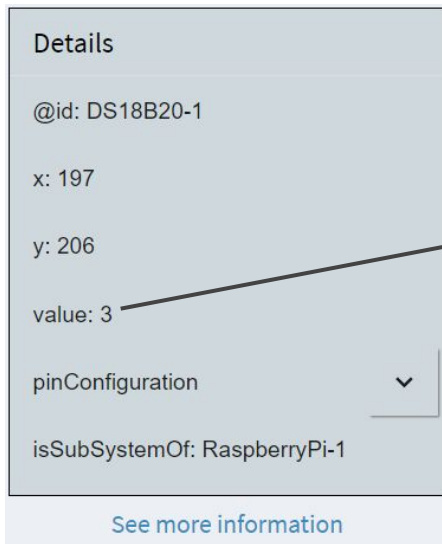
1. Devices
2. Sensors
3. Actuators

Each smart device/component has an id and an icon.



3.3.3 Information List

The information list contains information about the device/component, that is selected on at the moment.



The property **value** just shows up for sensing devices.

4. Roadmap

1. Separate user and admin view
2. Integration with hardware for getting information from a physical environment

