Private Public Partnership Project (PPP)

Large-scale Integrated Project (IP)

User & Programmer Manual

Author: AEON Cloud Messaging GE development team (javier.garcia@atos.net)

Contents

Cloud Messaging GE (AEON) Users and Programmers Guide

Introduction

The Cloud Messaging GE (AEON) has been thought to be a platform for developers. Therefore, the users and programmers guide will be explained the same way.

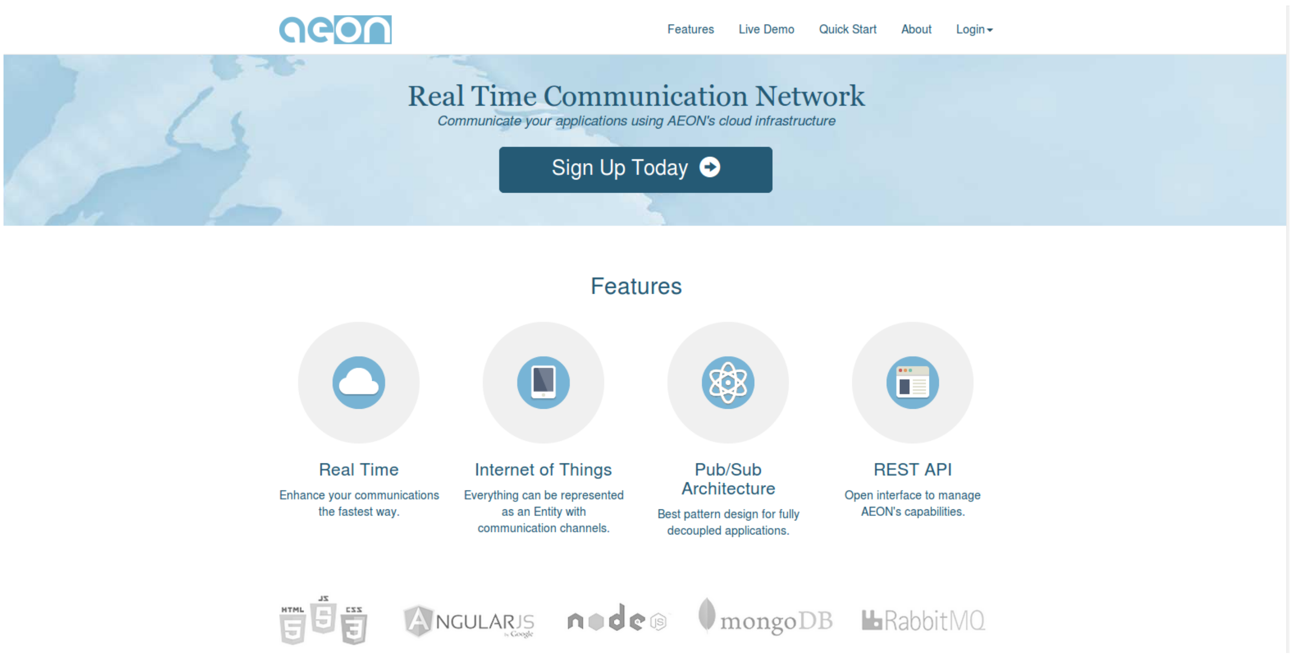
The Cloud Messaging GE provides a Dashboard in which it is possible to manage all the entities and their related channels in an easy way. It also provides several SDKs with the publish/subscribe operations ready to be used with pre-configured channels.

The current document explains both, the usage of the dashboard and the SDK.

To get a wider information about aeon, please, go to the README.md file.

Using AEON Dashboard

The image below shows the main page of the AEON Dashboard.



Real Time Communication Network

Communicate your applications using AEON's cloud infrastructure

Features

Real Time

Enhance your communications the fastest way.

Internet of Things

Everything can be represented as an Entity with communication channels.

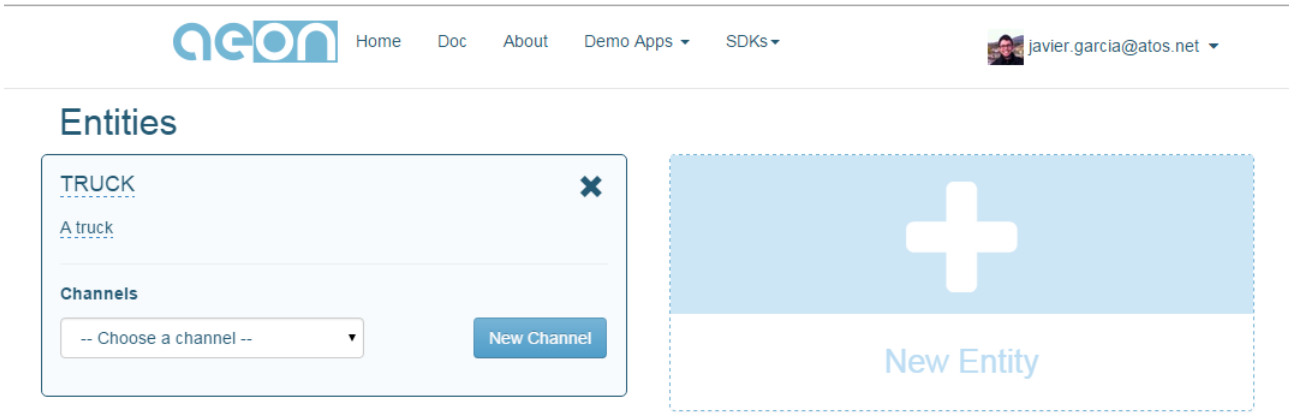
Pub/Sub Architecture

Best pattern design for fully decoupled applications.

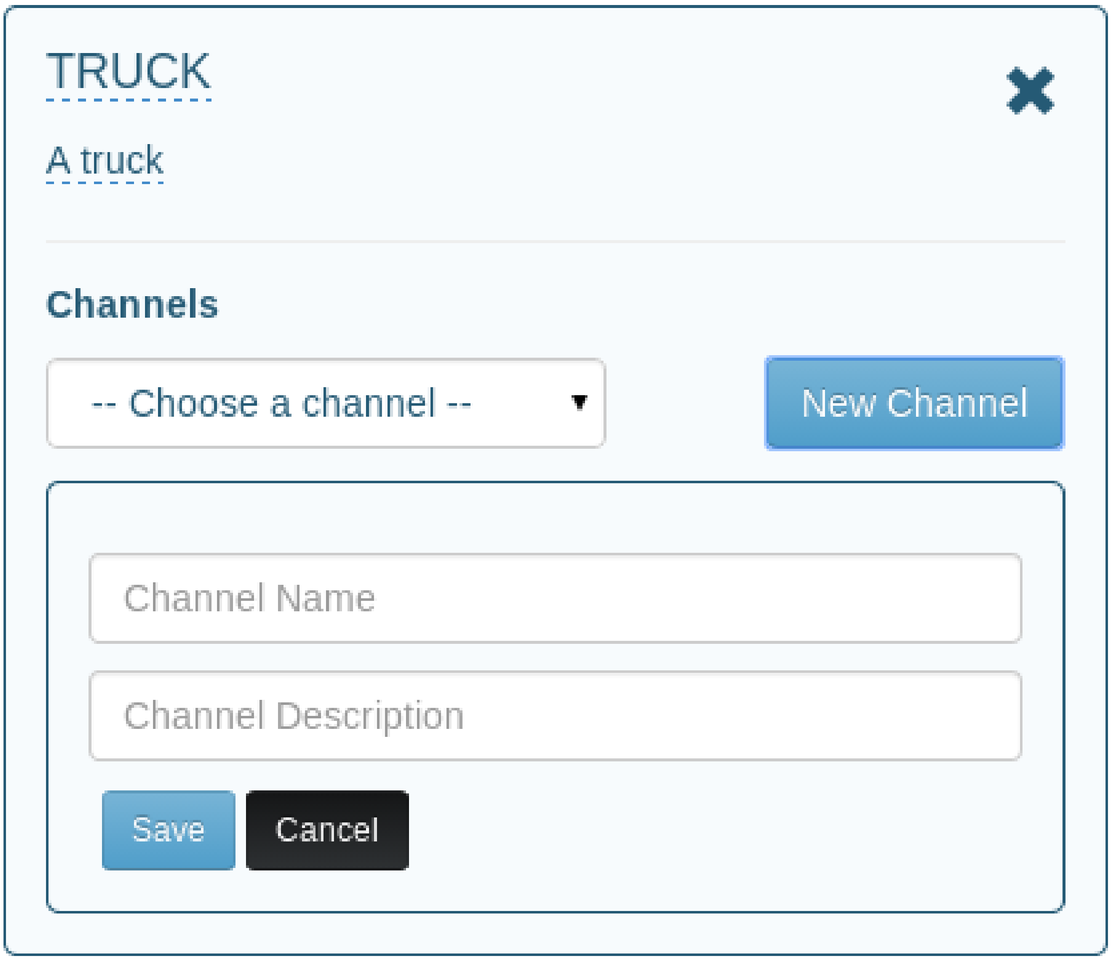
REST API

Open interface to manage AEON's capabilities.

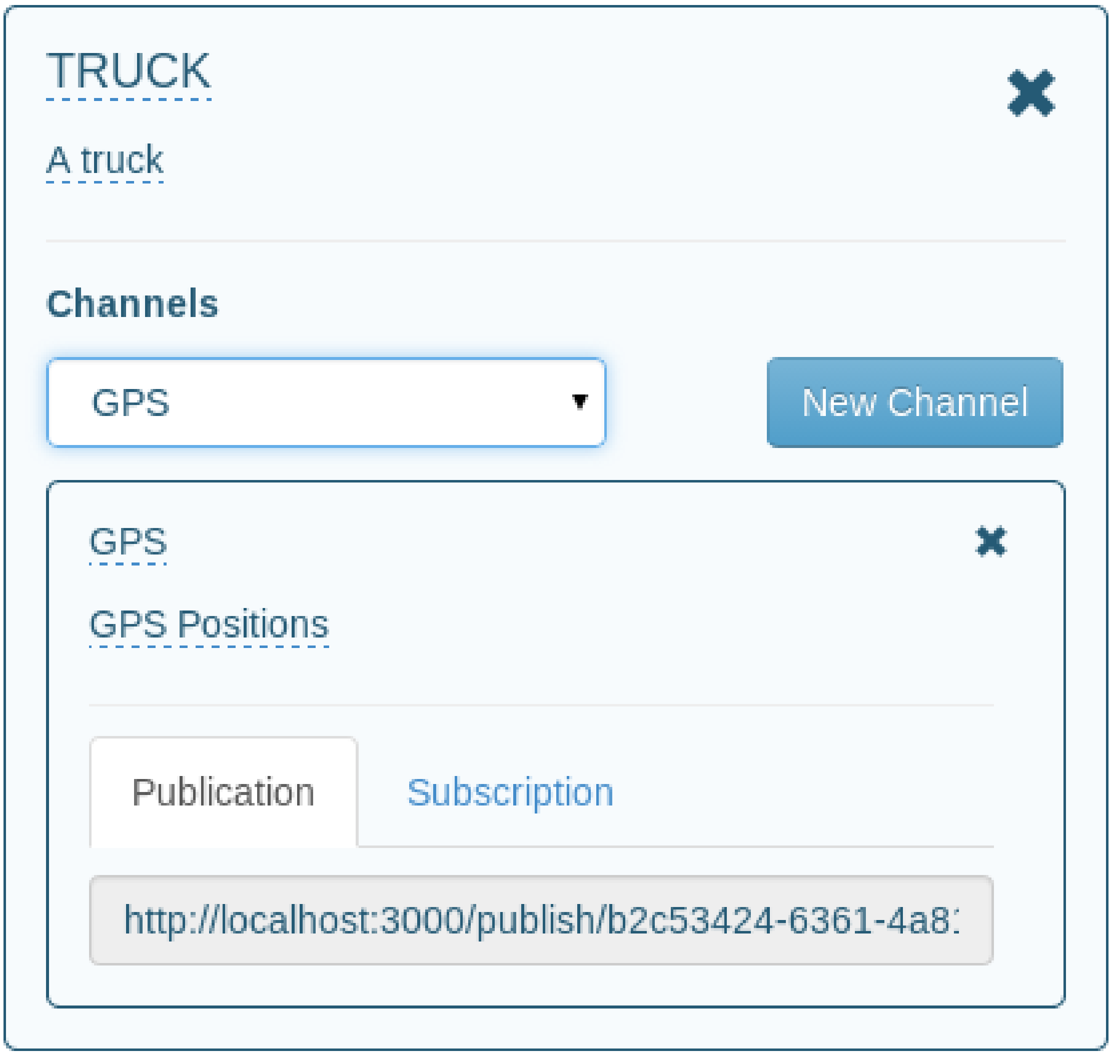
The user can sign in following the instructions or login if already registered. After login, the user will be able to see all the different entities previously created.



Each card belongs to a specific entity with its channels. Inside the card, there is a "New Channel" button in which, the user can create new communication channels for that specific entity.



When displaying the channel combobox, all the different channels will be shown to the user. Each of them contains two URLS: PUBLICATION and SUBSCRIPTION. These URLs will be used to publish the information or subscribe to a channel in order to get notification about other publishers on that channel.


TRUCK

A truck

Channels

GPS

New Channel

GPS

GPS Positions

Publication

Subscription

http://localhost:3000/publish/b2c53424-6361-4a8

To update the entity or the channel info, the user must click on the entity/channel name/description and an editable field will appear. The publish/subscribe URLs are not editable. To remove an entity/channel, the user only needs to press the cross next to the element.

Using AEON SDKs

AEON provides three different SDKs which help the developer to build their applications. The supported languages are: Java, Javascript and Node.js. The SDKs provide the publish/subscribe functionlaities. This means that, the user will need to get the PUB_URL and/or the SUB_URL by using the API or the Dashboard to get them.

To use the SDKs, the user needs to download it first. This can be done from the AEON's landing page: Cloud Messaging GE In the same page, it is possible to find some quickstart references to start using the SDK. Here, it is explained how to use it:

Hello World using Node.js (Publish)

Lets start with the simple code to publish some numbers. First of all you need to be connected to an AEON channel using your publish url.

First of all, it is necessary to instanciate the SDK and choose the operation mode. This will be obtained from the URL. In that sense, if the URL is for publishing, the SDK will be operating in Publish mode and it will only support operations related to the publication. On the other hand, it will operate in subcription mode.

```
var AeonSDK = require('aeonsdk-node');
sdk = new AeonSDK(config.PUB_URL); //The PUB_URL has been previously obtained
```

And now, lets generate some numbers with a delay of one second. To publish information you need to use a JSON object:

```

AEONSDK sdk = new AEONSDK(Config.PUB_URL);
for (var i=1; i < 20; i++){
   msg = { "number": i};
   sdk.publish(msg);
   sleep(2); //sync blocking sleep, ugly ;)
}
```

Done!Once connected to the channel, it is possible to directly invoke the publish method passing a JSON as a parameter.

Hello World using Node.js (Subscribe)

Now lets create the application to receive the numbers. Againg we need to create a connection with our previously created channel (this time, using the subscription url).

```
    var AeonSDK = require('aeonsdk-node');
    var subscriptionData = { "id": config.YOUR_ID, "desc": config.YOUR_DESC};
    sdk = new AeonSDK(config.SUB_URL, subscriptionData);
```

The combination of YOUR_ID and YOUR_DESC (strings) makes your subscriber unique. Once the SDK is ready you can ask for a subscription. As simple as:

```

    sdk = new AeonSDK(config.SUB_URL, subscriptionData);
    sdk.subscribe(received);
```

“received” parameter is callback function. This is how AEON manages asynchronous messages produced during the subscription. Any time a message is published in the channel; AEON will notify the user through the callback function.

Now lets do it a little bit more interesting. With AEON it is possible to pause the communication, messages produced during the break will be received when invoking the continue method:

```

    var received = function received(msg) {
        console.log("Received: ", msg)
    }

    sdk.subscribe(received);
    console.log("Ok, we are subscribed, waiting for messages");
    setTimeout(function(){
        sdk.pauseSubscription();
        console.log("lets pause 3 secs");

        setTimeout(function(){
            sdk.continueSubscription();
            console.log("lets continue");
            setTimeout(function(){
                sdk.deleteSubscription();
                console.log("Closing. Bye bye");
            }, 3000);
        }, 3000);
    }, 3000);
```

When everything is finished you can delete the subscription. After that, if you invoke again sdk.subscribe() with your ID and DESC you will receive a new one, instead of having the possibility of continuing with your previous subscription. Dont delete the subscription if you want to use it later (tomorrow, the next month, in the year 2024).

Now, the user should be able to play with this create, pause, continue, delete depending on the logic of his application. Happy coding.

To get a more detaile information about the SDK, please, follow the link: SDK for Node.js