Operations with Service Bus

1. Introduction

“Data is transferred between different applications and services using messages. A message is a container decorated with metadata. The data can be any kind of information encoded in any format such as: JSON, XML, Plain Text, etc.”-Microsoft Docs. Service buses allow us to micromanage our applications and allow for certain applications to get the information they need. Here I will describe how the service bus helps us to receive any incoming messages, filter them and save them to our database.

2. Working with Service Bus

I have created a web application called ‘dev-servicebus’ which is listening for new incoming messages on the service bus. When a new message arrives the attributes from data package are extracted and checks if the device name contains a specific tag, if it does then the information is saved to PostgreSQL database. It creates a new table with the name that was specified in the application and stores the samples of the package inside that table. To avoid data latency, the other specific properties of the package are stored inside a separate table called ‘data_properties’ and it is connected to the samples table using a foreign key. To get the information of the samples table with its corresponding configuration of data properties one must use an inner join while querying the data in the PostgreSQL portal. The query should look like this:

```
SELECT *
FROM <table_name>
INNER JOIN data_properties
   ON data_properties.id = <table_name>.property_id
WHERE tstamp >= 'yyyy-mm-dd hh:mm:ss' AND tstamp <= 'yyyy-mm-dd hh:mm:ss';
```

The ‘where’ clause is not required, it is included in case we want to query data that is in a specific time range. It is important to note that for the data to be saved it should include the ‘dsr’ tag in the device id name before connecting to the transmitter.