



Instituto Superior Técnico

Production Management

# **ThingsBoard Guide**

Marta Batalha and Ricardo Febra

September 27, 2022



# Contents

<b>1</b>	<b>Introduction</b>	3
<b>2</b>	<b>Device</b>	4
2.1	Create a device.	4
2.2	Delete a device.	5
2.3	Rename a device	5
2.4	Change the device profile of a device	5
2.5	Check device last telemetry	6
<b>3</b>	<b>Device Profile</b>	7
3.1	Create a device profile.	7
3.2	Rename or delete a device profile	8
3.3	Change the rule chain of a device profile	8
<b>4</b>	<b>Rule Chain</b>	8
4.1	Root Rule Chain	9
4.2	Create a new rule chain	10
4.3	Rename or delete a rule chain	11
4.4	Export and import a rule chain.	11
4.5	Example	12
<b>5</b>	<b>Dashboard</b>	15
5.1	Create a dashboard	16
5.2	Create an alias	16
5.3	Add a new widget	17
5.4	Edit data timewindow	19
5.5	Rename or delete a dashboard	19
5.6	Export and import a dashboard.	20
5.7	Example	20
5.7.1	Simple card widget	20
5.7.2	Label widget	22
5.7.3	Timeseries table widget	23
5.7.4	Timeseries chart widget	24
<b>6</b>	<b>Further information</b>	25



## 1 Introduction

ThingsBoard is an open-source IoT platform for data collection, processing, visualization, and device management. It allows to represent complex physical world systems via assets, devices and relations. Those devices provide data that can be processed and used to easily and quickly produce more valuable insights. It is then possible to design real time dashboards to visualize the data, using a simple drag-and-drop editor interface that contains a vast widget library.

This document explains some basic features of ThingsBoard platform. The goal is to provide support to Production Management students when developing their projects. It covers topics such as the creation of devices, device profiles, rule chains and dashboards. It provides guidelines with step-by-step information and shows some examples.



## 2 Device

To represent the physical devices used in the laboratory, ThingsBoard devices are created. These may belong to different device profiles. In addition, it is possible to have different root rule chains governing different device profiles. These concepts will be better explained in further topics.

### 2.1 Create a device

A device in ThingsBoard platform should be added for each physical device being used in the laboratory (iStartLab V2.0 board, receiver board and vibration sensor board). To do that, first select Devices from ThingsBoard menu (Figure 1).

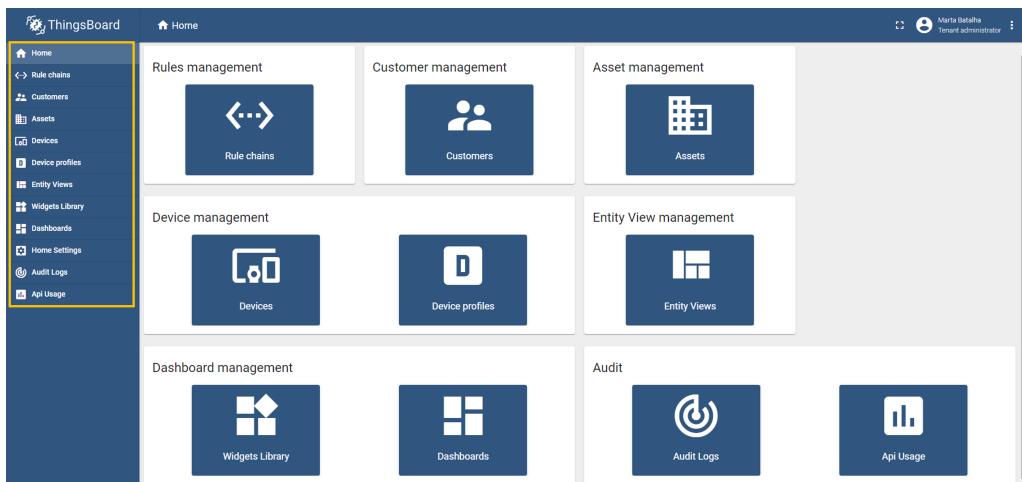


Figure 1: ThingsBoard home page, with its menu highlighted in yellow

Then, on the top right corner, click on "+" to add a new device and a new window shows up. To add the device, follow the instructions in Figure 2. Once these steps are fulfilled, a new device is created and added to the devices list. This list can be accessed by simply going to Devices, on ThingsBoard menu.



The screenshot shows the 'Add new device' dialog box. It has three main sections: 1. Device details (highlighted with a yellow circle containing the number 1), which includes fields for Name (Example Device), Label, Transport type (Default), Device profile (default), and options for Create new device profile and Is gateway. 2. Credentials (Optional). 3. Customer (Optional). At the bottom right, there are buttons for Cancel, Add (highlighted with a yellow circle containing the number 2), and Next: Credentials.

Figure 2: Add a new device

## 2.2 Delete a device

To delete a device (Figure 3), find the desired device on the devices list and click on the trash symbol.



Figure 3: Delete a device

## 2.3 Rename a device

To rename a device, it is necessary to access the device details. To do so, find the desired device on the devices list and click once on the top of it. Next, follow the instructions in Figure 4, with step 2.1 representing the renaming process.

## 2.4 Change the device profile of a device

To change the device profile of an existing device, it is also necessary to access the device details, as in for the renaming process. Selecting the



desired device profile is expressed in step 2.2 of Figure 4. Despite the two processes being represented in the same figure, they can be done separately.

Note that, for this procedure, it is necessary to first create the device profile (see section 3.1 of this document).

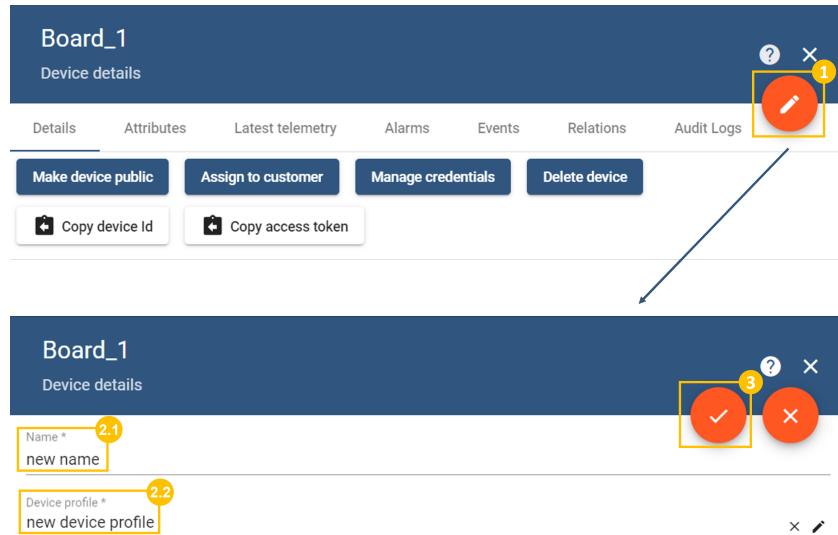


Figure 4: Rename a device (2.1) and change the device profile (2.2)

## 2.5 Check device last telemetry

Last telemetry consists on a list of all the keys (variables) associated to a device. For each of these keys, it contains the value from the last message sent by the device, as well as the correspondent timestamp. To check it, go to device details and select the Last telemetry tab (Figure 5). There is a particular circumstance where it is very useful to consult Last telemetry. When an experiment is being setup, by verifying the timestamps, it is possible to understand if the device is receiving real time data, allowing to detect communication related issues.



The screenshot shows the ThingsBoard interface for a device named 'Board\_1'. The top navigation bar includes tabs for 'Details', 'Attributes', 'Latest telemetry' (which is highlighted with a yellow border), 'Alarms', 'Events', 'Relations', and 'Audit Logs'. There are also icons for help, close, and edit. Below the navigation is a table titled 'Latest telemetry' with three rows of data:

Last update time	Key ↑	Value
2022-01-20 19:21:00	blueLDR	16
2022-01-20 19:21:01	brownLDR	31

Figure 5: Last telemetry

## 3 Device Profile

### 3.1 Create a device profile

In ThingsBoard, every device belongs to a device profile. This means a device profile can have multiple devices associated, but each device belongs to one and only one device profile at a single point in time.

ThingsBoard automatically creates a default device profile and, if nothing is specified, every new created device is assigned to this profile. When a new device profile is created, it is possible to configure some settings, which, when working with several different devices at the same time, might be really useful.

The procedure used to create a new device profile is analogous to the one used to add a new device. First, select Device profiles from ThingsBoard menu. Then, on the top right corner, click on "+" to create a new device profile and a new window shows up. To create the device profile, follow the instructions in Figure 6. Once these steps are fulfilled, refresh your web page and a new device profile was created and added to the device profiles list. This list can be accessed by simply going to Device profiles, on ThingsBoard menu.



The screenshot shows the 'Add device profile' dialog box. At the top, there are four tabs: 1 Device profile details, 2 Transport configuration (Optional), 3 Alarm rules (0) (Optional), and 4 Device provisioning (Optional). The first tab is selected. Below the tabs, there is a 'Name \*' input field containing 'Example Device Profile', which is highlighted with a yellow border and a circled '1'. Below the input fields are sections for 'Rule chain', 'Queue Name', and 'Description'. At the bottom right of the dialog is a 'Next: Transport configuration' button, a 'Cancel' button, and a prominent 'Add' button, which is also highlighted with a yellow border and a circled '2'.

Figure 6: Create a new device profile

### 3.2 Rename or delete a device profile

To delete or rename a device profile are similar processes to the ones of deleting and renaming a device. Therefore, it is suggested to consult sections 2.2 and 2.3 of this document.

### 3.3 Change the rule chain of a device profile

To change the rule chain of an existing device profile, it is necessary to access the device profile details. To do so, find the desired device profile on the list and click once on the top of it. Next, follow the instructions in Figure 7. Note that, for this procedure, it necessary to first create the rule chain (see section 4.2 of this document).

## 4 Rule Chain

Rule chains are part of the ThingsBoard rule engine, an easy to use framework for building event-based workflows. Messages (any incoming event) are received, and then passed through rule nodes (a function that is executed on an incoming message), which all connected with relations result in a rule chain. The rule nodes can be specially important if one wants to create new keys based on the analysis of keys belonging to the incoming messages.

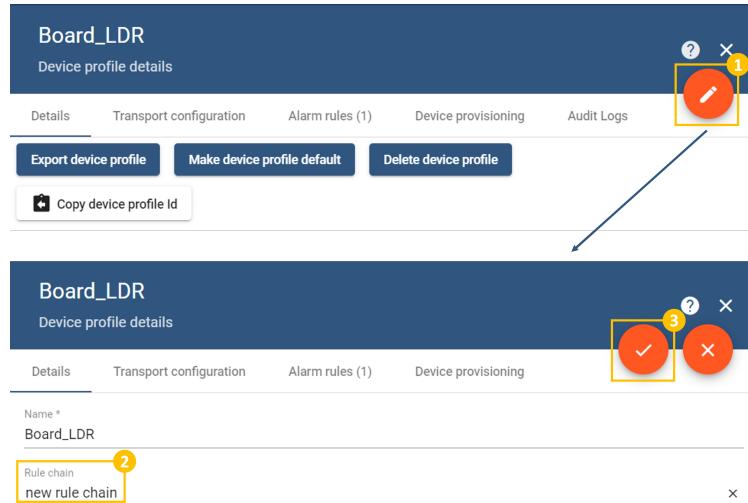


Figure 7: Change the rule chain of a device profile

#### 4.1 Root Rule Chain

As for device profiles, ThingsBoard automatically creates a default rule chain, which will be assigned to every new device profile, if nothing is specified. It is called Root Rule Chain (Figure 8).

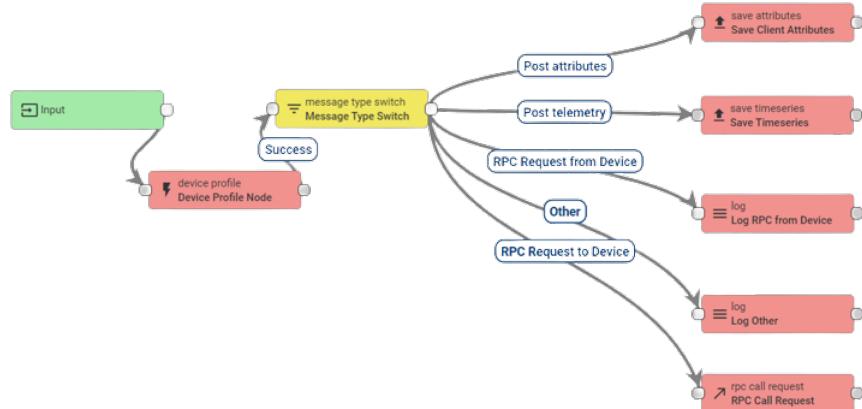


Figure 8: Root Rule Chain

When a device receives a message, its content goes through all the nodes and relations in the Root Rule Chain. It is important to highlight the "Save Timeseries" node. This node stores timeseries data from the incoming



message content to the database. In case any further data processing is required, other nodes and relations will link to this node.

Also note that, by default, the Root Rule Chain processes all incoming messages and events for any device. However, different device profiles might require different data processing. It is recommended not to use the default Root Rule Chain, so it works as a backup rule chain. Instead, the suggestion is to create a new root rule chain for each device profile, where smaller rule chains can then be invoked.

## 4.2 Create a new rule chain

To create a new rule chain, first select Rule Chains from ThingsBoard menu. Then, on the top right corner, click on "+" to create a new rule chain and a new window shows up. Give it a name and click on "Add". Once these steps are fulfilled, a new rule chain was created and added to the rule chains list. This list can be accessed by simply going to Rule chains, on ThingsBoard menu.

To open a rule chain, first access its details, by finding the desired rule chain on the list and clicking once on the top of it. Then, click on "Open rule chain" (Figure 9).

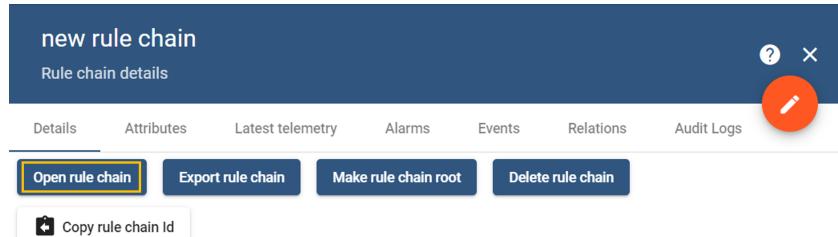


Figure 9: Open rule chain

Once a new rule chain is opened, it only has an input node. It is then possible to edit and build it, according to specific problem requirements, using the simple drag-and-drop editor interface. In the left side, there is a drop down menu with all the available nodes (Figure 10). To see some information about each node, just hover the mouse over it and a brief explanation shows up.

A very important remark: whenever a change is performed, it has to be saved by clicking in the orange circle with the tick symbol, in the bottom



right corner (Figure 10).

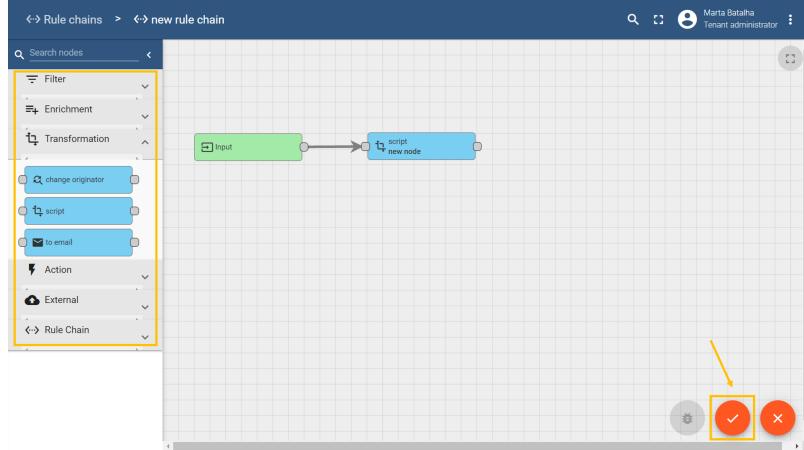


Figure 10: Edit a rule chain

### 4.3 Rename or delete a rule chain

Delete or rename a rule chain are similar processes to the ones of deleting and renaming a device. Therefore, it is suggested to consult sections 2.2 and 2.3 of this document.

### 4.4 Export and import a rule chain

It is possible to export a rule chain (Figure 11). When this is done, a .json file is downloaded to the computer. With that file, if later one wants, it is possible to import the rule chain, by clicking on "+" on the top right corner, as shown in Figure 12.

Figure 11: Export rule chain

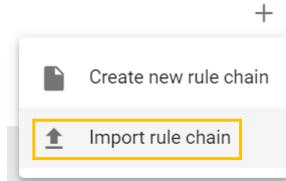


Figure 12: Import rule chain

## 4.5 Example

In this section, a practical example with the relation between device profiles, root rule chains and rule chains, is described.

There is a physical iStartLab board equipped with a Time of Flight (ToF) sensor. With the goal of detecting the presence of a person, it will be installed in the edge of a table, pointing forward (this is part of demonstration D). The message sent to ThingsBoard contains the key *msg.data.analogInput\_4*, whose values gives the measured distance to the first obstacle, in centimeters. However, it is more convenient to have a key that assumes values in/out, according to the measured distance. The adopted procedure to tackle this situation is presented bellow.

### Create a rule chain to assess the presence of a person

A new rule chain called "ToF D - Worker Presence" was created. Two nodes were added to the input one and connected between each other, according to Figure 13.



Figure 13: Rule chain "ToF D - Worker Presence"

The blue node is a script transformation node and allows the transformation of the message payload through a JavaScript function, written by the user. Double-click on the node to access its details, like the JavaScript code used (Figure 14). For this example it was defined that, if the measured distance is greater than 1 meter, it should be considered that there is no person at the workbench. A simple if-else statement was used and in either case, a variable called *workerState* receives a new value, IN or OUT according to the defined threshold.



```
function Transform(msg, metadata, msgType) {
    1- if (msg.data_analogInput_4 > 100){ //Define distance to workbench
    2-     msg.workerState = "OUT";
    3- }
    4- else if (msg.data_analogInput_4 <= 100){
    5-     msg.workerState = "IN";
    6- }
    7-
    8 return {msg: msg, metadata: metadata, msgType: msgType};
```

Figure 14: JavaScript code in "Worker Presence" script node

For the reasons expressed previously, since the message payload is undergoing changes, a "Save Timeseries" node must be added. The configuration of this node is very simple (Figure 15). When linking the last two nodes, a "Success" label should be selected.

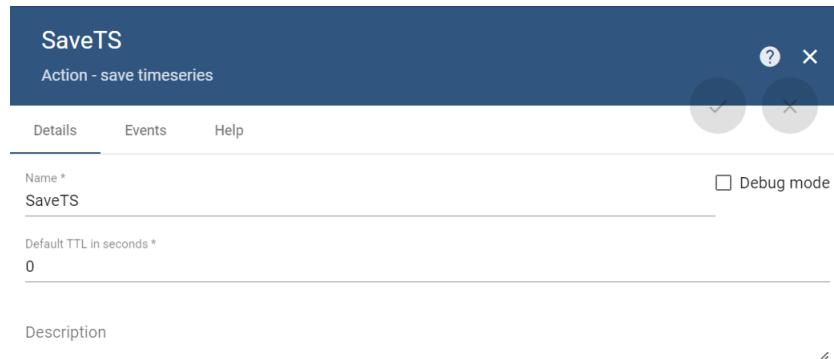


Figure 15: Save timeseries node

### Create a new root rule chain

A new root rule chain called "ToF D - Root Rule Chain" was created. First, all nodes and relations in "Root Rule Chain" (the default one, created by ThingsBoard) were copied and added to the input node. Then, a rule



chain node (the purple one) was added to the "Save Timeseries" one. This node enables to forward incoming messages to specified rule chains, in this example the "ToF D - Worker Presence" rule chain. The final result is shown in Figure 16.

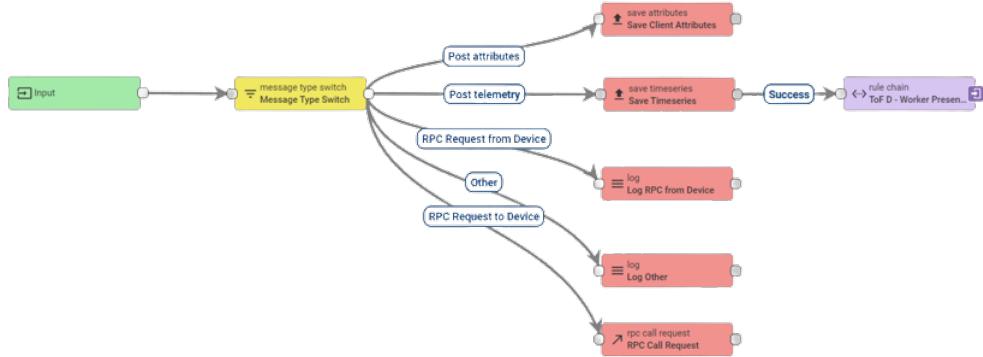


Figure 16: Root rule chain "ToF D - Root Rule Chain"

### Create a new device profile

A new device profile named "Board\_ToF\_D" was created. In its details, "ToF D - Root Rule Chain" was the choice for its rule chain (Figure 17).

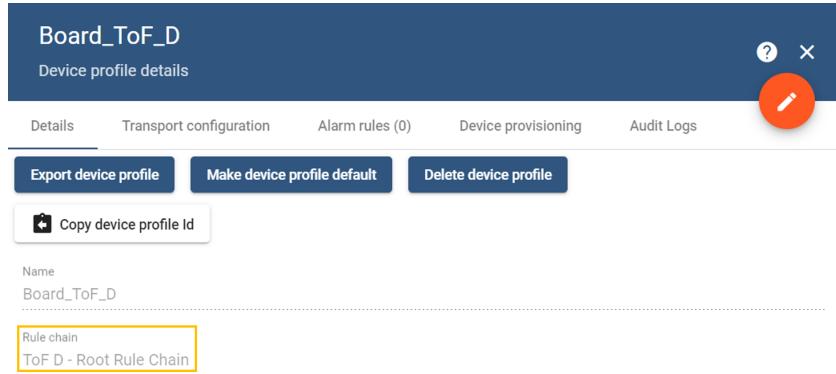


Figure 17: Device profile "Board\_ToF\_D" details

### Change the device profile of the device

"Board\_8" is the name of the device receiving messages with measured distance. Its device profile was changed to "Board\_ToF\_D" (Figure 18).



The screenshot shows the 'Device details' page for 'Board\_8'. At the top, there are tabs for 'Details', 'Attributes', 'Latest telemetry', 'Alarms', 'Events', 'Relations', and 'Audit Logs'. Below these are several buttons: 'Make device public', 'Assign to customer', 'Manage credentials', and 'Delete device'. There are also two links: 'Copy device Id' and 'Copy access token'. The 'Name' field contains 'Board\_8'. A 'Device profile' dropdown menu is open, showing 'Board\_ToF\_D' which is highlighted with a yellow box. At the bottom right of the page is a red circular button with a white pencil icon.

Figure 18: Device "Board\_8" details

### Check the last telemetry of the device

After all these steps, once the device is setup in the laboratory and starts collecting data again, the variable *workerState* will show up in "Board\_8" last telemetry (Figure 19).

The screenshot shows the 'Latest telemetry' page for 'Board\_8'. At the top, there are tabs for 'Details', 'Attributes', 'Latest telemetry', 'Alarms', 'Events', 'Relations', and 'Audit Logs'. The 'Latest telemetry' tab is selected. Below it is a table with columns: 'Last update time', 'Key ↑', and 'Value'. The first row shows '2021-12-15 12:32:39' and 'workerState'. The second row shows 'OUT'. The 'workerState' entry is highlighted with a yellow box. At the bottom right of the page is a red circular button with a white pencil icon.

Figure 19: "Board\_8" last telemetry

## 5 Dashboard

ThingsBoard provides the ability to create and manage dashboards, which are useful to visualize real time data from many different devices. There are plenty of widgets that allow to show information in multiple ways.



## 5.1 Create a dashboard

To create a new dashboard, first select Dashboards from ThingsBoard menu. Then, on the top right corner, click on "+" to create a new dashboard and a new window shows up. Give it a name and click on "Add". Once these steps are fulfilled, a new dashboard was created and added to the dashboards list. This list can be accessed by simply going to Dashboards, on ThingsBoard menu.

To open a dashboard, first access its details, by finding the desired dashboard on the list and clicking once on the top of it. Then, click on "Open dashboard" (Figure 20).

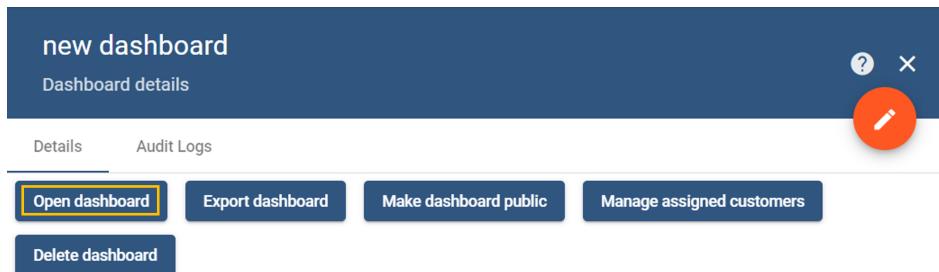


Figure 20: Open dashboard

## 5.2 Create an alias

To add data to a dashboard, it is necessary to create aliases to define to which entities (that can be devices) belong the data to be shown. To create an alias, first open the desired dashboard, and then follow the steps shown in Figure 21. For the configuration of the new alias, see Figure 22. Note that, in the field "Device", all the devices in the device list will show in a drop-down-list. Choose the device that contains the data to be shown in the dashboard. A final remark goes to the fact that, if one wants to show data from two devices, two aliases should be added, one for each device. It is possible to add as many aliases as wanted.

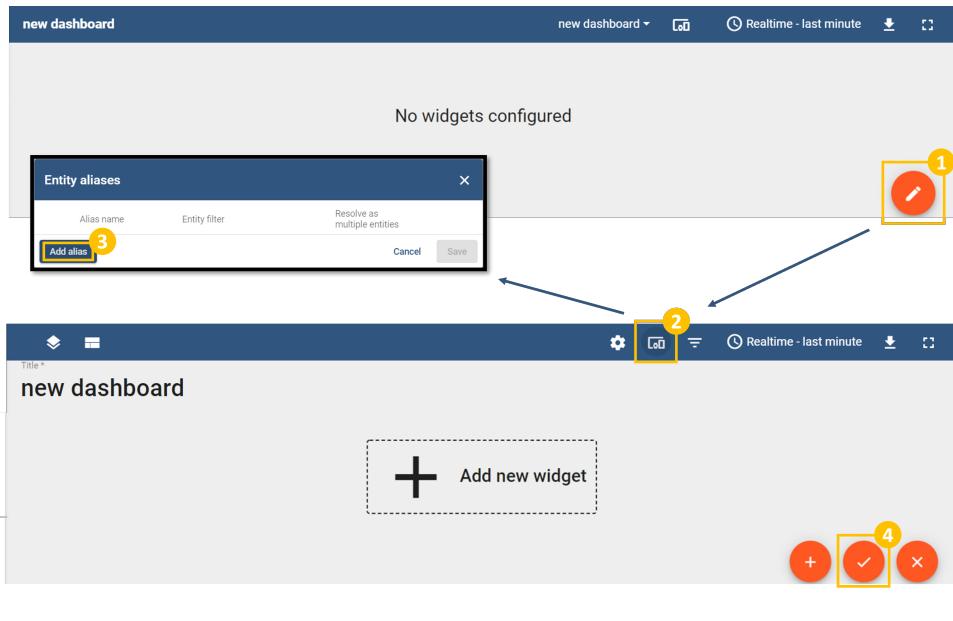


Figure 21: Add alias

The dialog is titled 'Add alias'. It contains the following fields:

- 'Alias name \*': A text input field with 'new alias' typed.
- 'Resolve as multiple entities': A toggle switch that is off.
- 'Filter type \*': A dropdown menu set to 'Single entity'.
- 'Type \*': A dropdown menu set to 'Device'.
- 'Device \*': A dropdown menu set to 'new device'.

At the bottom right are 'Cancel' and 'Add' buttons, with 'Add' being highlighted by a yellow square.

Figure 22: New alias

### 5.3 Add a new widget

Once aliases are created, it is possible to add new widgets. First open the desired dashboard and then follow the steps shown in Figure 23.

Then, a widget library will open up. Search and choose the most appro-

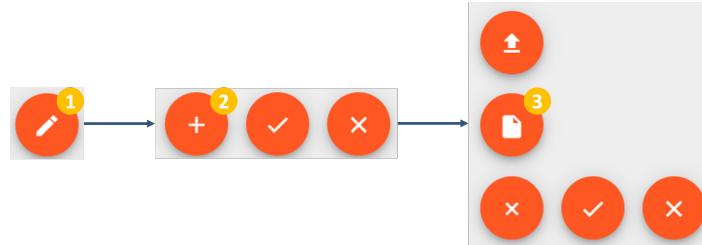


Figure 23: Add new widget

priate widget type. Regardless the chosen widget, the next step will always be to choose the datasource (Figure 24). In the blue field, choose one of all aliases. On the other hand, in the green field, choose the keys (associated with the chosen alias) to be shown in that widget.

Figure 24: Choose datasource

The widget configurations (title, text color, text size, background color, etc.) can be changed anytime. To do so, just enter in dashboard edit mode, then click on the pencil symbol in the widget to be edited and go to "Settings" tab. In section 5.7 of this document, some examples can be found.

A very important remark: whenever a change is performed, it has to be saved by clicking in the orange circle with the tick symbol.



## 5.4 Edit data timewindow

One of the great advantages of ThingsBoard platform is the possibility to have real time dashboards to consult. However, in some situations, it might be useful to check data from a specific time interval. Fortunately, this is also possible.

To do it, first open the desired dashboard and then click on the clock symbol, in the top right corner (Figure 25).



Figure 25: Access timewindow edit mode

Then it is possible to choose "Realtime" (Figure 26a) or "History" (Figure 26b).

(a)

(b)

Figure 26: Edit timewindow

## 5.5 Rename or delete a dashboard

Delete or rename a dashboard are similar processes to the ones of deleting and renaming a device. Therefore, it is suggested to consult sections 2.2 and 2.3 of this document.



## 5.6 Export and import a dashboard

Export and import a dashboard (or a widget) are similar processes to the ones of exporting and importing a rule chain. Therefore, it is suggested to consult section 4.4 of this document.

## 5.7 Example

### 5.7.1 Simple card widget

This widget is ideal to use when one wants to show the current value of a key.

In Figure 27, three examples of simple card widgets are presented. The one in Figure 27a shows a state, while the ones in Figures 27b and 27c show time and distance measures, respectively.

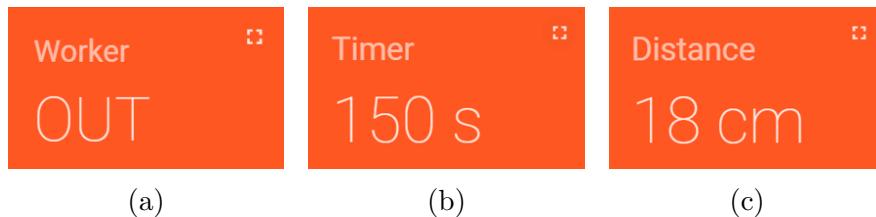


Figure 27: Simple card widget examples

Looking at example 27a. In order to have "Worker" written on the top left corner of the widget, the label of the key to be shown has to be changed. To do so, when setting the datasource (Figure 28), click on the pencil symbol next to the key name. A new window opens, where the field "Label" can be changed (Figure 29).

The image shows a screenshot of a configuration interface for a widget's datasource. At the top, there is a header 'Datasources' with a note 'Maximum 1 datasource is allowed.' Below this, there are two columns: 'Type' and 'Parameters'. Under 'Type', there is a dropdown menu set to 'Entity'. Under 'Parameters', there is a row with 'Entity alias\*' followed by the value 'Board\_8'. To the right of this row is a small edit icon (pencil symbol) and a delete icon (cross symbol). A yellow arrow points from the text 'Widget datasource' in the previous figure to this edit icon. Below the parameters section, there is a note 'Maximum 1 timeseries/attribute is allowed.'

Figure 28: Widget datasource



Data key configuration X

Key \*  
workerState X

Label \*  
Worker Color \*

Special symbol to show next to value Number of digits after floating point

Use data post-processing function

Figure 29: Change key label

For this example, the widget settings are shown in Figure 30. A special remark for the field named "Special symbol to show next to value". In this case it does not make sense to use it. But, for instance, if a time or distance value is being displayed (like in examples 27b and 27c), it can be important to use this field to identify the proper units.

Worker state  
Simple card

?

Actions

Data      Settings      Advanced

General settings

Title  
Worker state

Title Tooltip

Display title icon  Icon Icon color  
rgba(0, 0, 0, 0.87) Icon size  
24px

Display title  Drop shadow  Enable fullscreen

Widget style \*  
1 [ ] Tidy Mini [ ]

Background color Background color  
#ff5722

Text color Text color  
rgba(255, 255, 255, 0.87)

Padding 16px

Margin Margin

Special symbol to show next to value Number of digits after floating point  
0

Display legend Legend settings [ ]

Mobile mode settings

Figure 30: Widget settings

### 5.7.2 Label widget

This widget is similar to the simple card widget. The main difference is that this one allows to display the key value over a background image.

Figure 31 presents an example where the displayed key indicates a tool availability state, while an image of that tool is used as the widget background.



Figure 31: Label widget example

To add the image as background, it is necessary to go to the widget "Advanced" tab (Figure 32). In the field "Pattern", it is defined how the key value is presented. For this example, it is simply displaying the key value, without any additional text. The values in fields X and Y "Percentage relative to background" can be modified, to alter the position of the image.

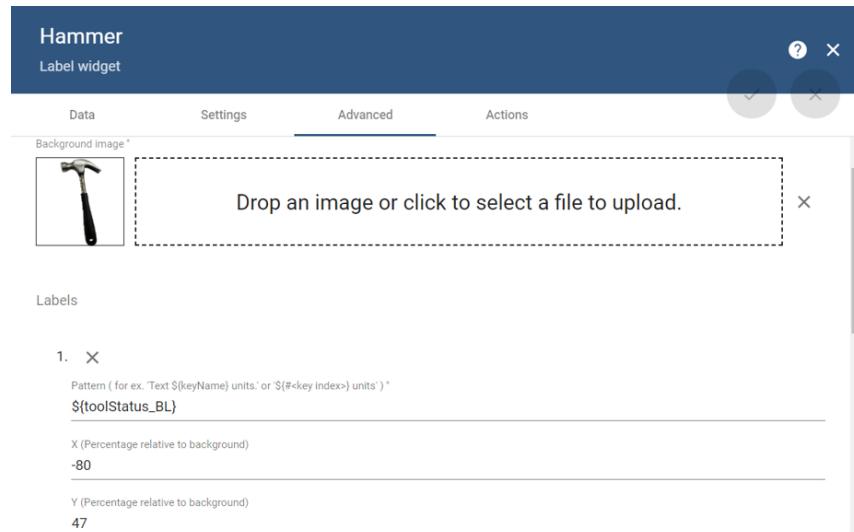


Figure 32: Widget advanced settings



### 5.7.3 Timeseries table widget

With this widget it is possible to visualize data in a table format. The first column indicates the timestamp of each row, while every row corresponds to a different message payload. The remaining columns show the value of the chosen keys.

In Figure 33, for every message (with its own timestamp), the values of two keys ("ID" and "Value") are presented. The choice of these keys is done when defining the datasource of the widget (Figure 34).

LDR percentage values					
🕒 Realtime - last minute					
Timestamp ↓	ID	Value			
2021-12-14 12:29:54	4	29			
2021-12-14 12:29:52	4	14			
2021-12-14 12:29:52	3	6			

1 – 10 of 200   

Items per page: 10

Figure 33: Timeseries table widget example

Datasources

Type	Parameters
Entity	<p>Entity alias * Placa_1</p> <p>=  ↗ ID: data_analogInput_0  </p> <p>=  ↗ Value: data_analogInput_1  </p> <p>Filter</p>

Figure 34: Widget datasource

This widget has an interesting feature that allows to format a row style



according to the key values it contains. Consider a different example (Figure 35), where two keys, one per task, can assume one of two values, "success" or "failed".

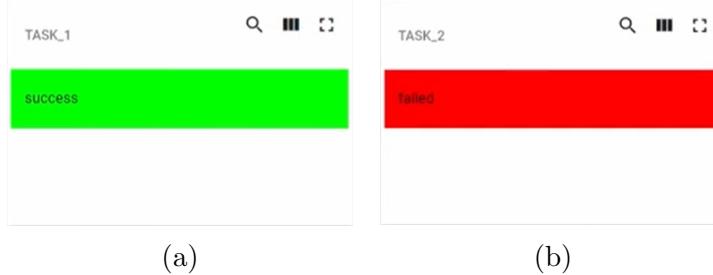


Figure 35: Row style function example

In "Advanced" widget tab, select the last option ("Use row style function"). Then, write a JavaScript function with the condition for row style format. Figure 36 shows the code used for this example. Marked with a yellow rectangle, there is the name of the key to consider for format condition, while in blue there is the desired value to trigger a specific style format. The colors should be indicated using hex color codes. For example, the green code (marked with a green rectangle) is #00FF00.

```
1 result = {background:"white"};
2 if (entity.taskState_1 == "Success") {
3     result.background = '#00FF00'; //green
4 } else {
5     result.background = '#FF0000'; //red
6 }
7 return result;
```

Figure 36: Widget datasource

#### 5.7.4 Timeseries chart widget

This is a simple widget with great interest when it is pretended to show a key value variation with time.

In the example of Figure 37, the evolution of the distance measured by a Time of Flight sensor is shown.

It is possible to show multiple keys in the same chart. For that, select all the desired keys when defining the widget datasource.

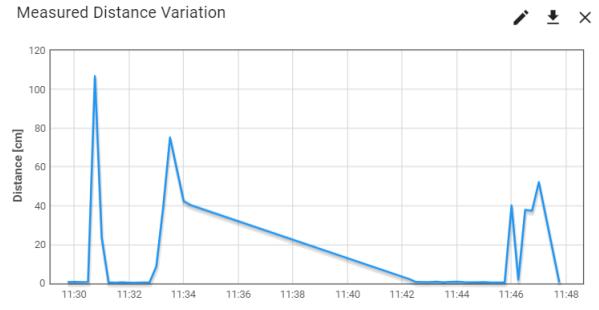


Figure 37: Timeseries chart widget example

## 6 Further information

This document only contains the basic concepts and procedures required to develop a simple project. ThingsBoard has much more features that can be explored.

For that, it is suggested the consultation of ThingsBoard website, which has many tutorials with step-by-step instructions (available in this [link](#)).

In addition, ThingsBoard YouTube channel (available in this [link](#)) has some tutorial videos with application examples.