**What Are *Alarms* in ThingsBoard?**

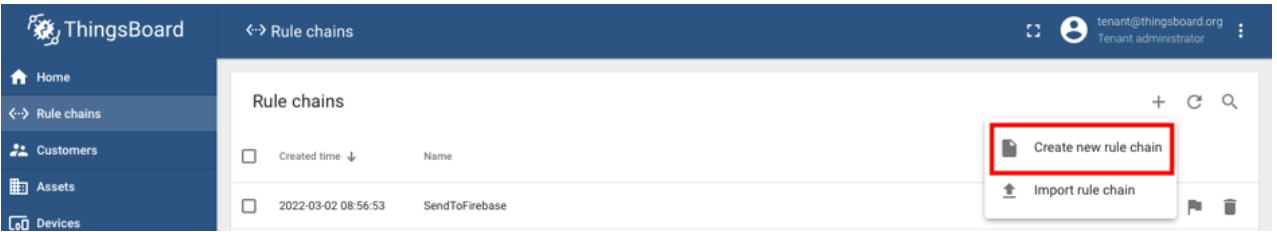
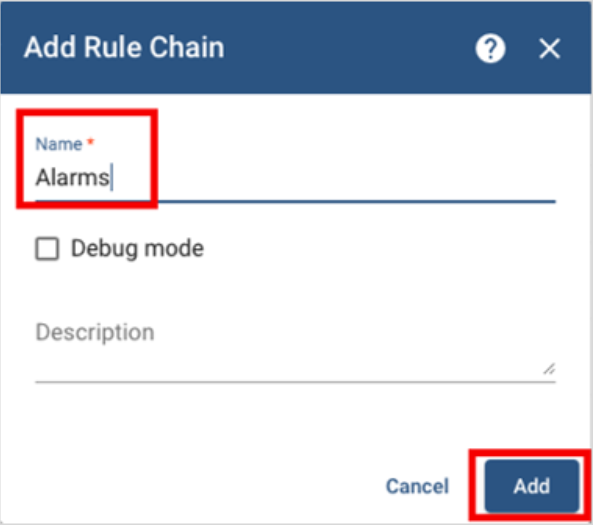
Imagine that you have an IoT device, such as a thermostat, that sends temperature data to ThingsBoard. Ideally, you would like to monitor your data so that if the temperature gets too high or too low, you will receive a notification. To achieve this, ThingsBoard provides the option to set *alarms* to monitor data and notify users or *clients* when the data reaches a certain threshold.

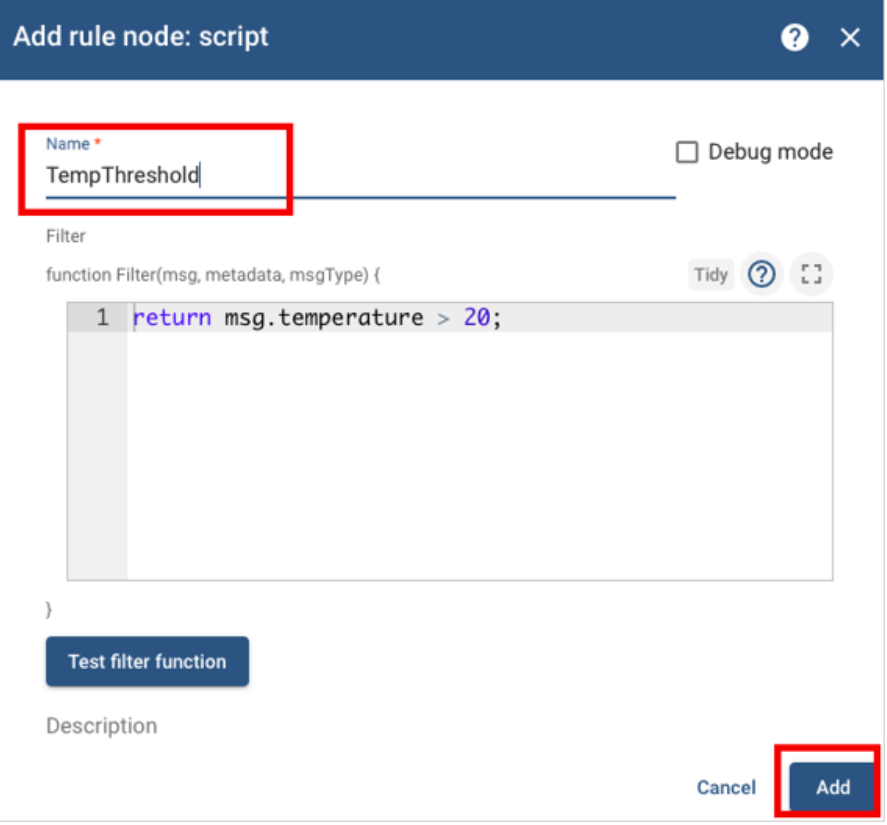
**How Are *Alarms* Created in ThingsBoard?**

The easiest way to create an *alarm* in ThingsBoard is by creating a custom rule chain that can trigger or clear your *alarms* based on your data readings.

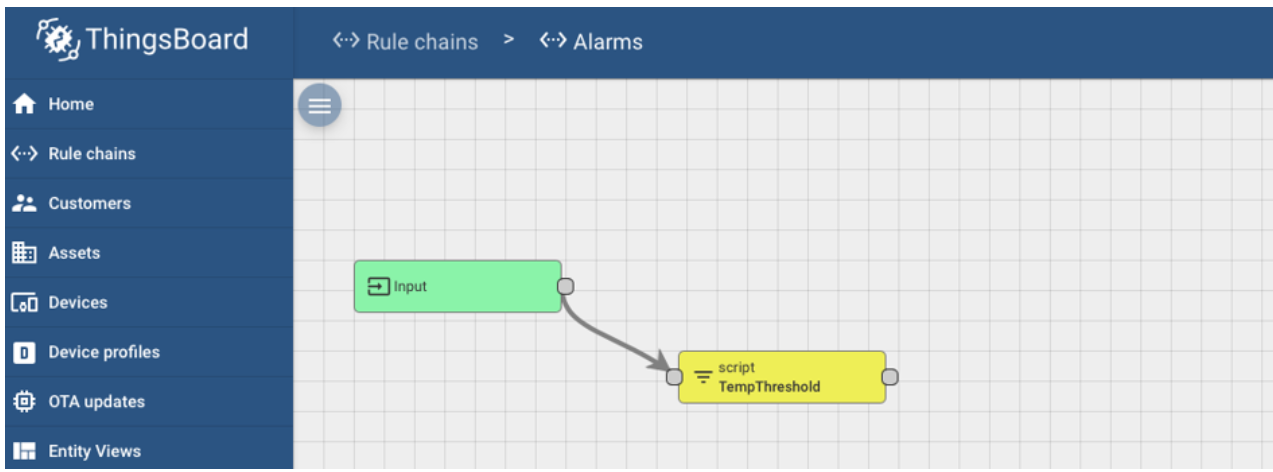
In this mini-lesson, you will learn how to create a custom rule chain to set up *alarms* in ThingsBoard. Then, you will learn how to connect *alarms* to your Root Rule Chain in [Videos 24.10 and 24.11](https://classroom.emeritus.org/courses/10605/pages/setting-up-alarms-in-thingsboard-to-monitor-data-10-23).

To set up an *alarm* rule chain, navigate to Rule Chains on the left-hand side menu of the ThingsBoard home page. Add a new rule by selecting the plus sign (+) icon in the top-right corner and then selecting “Create new rule chain”.

Name this rule chain “*Alarms*”, then select “Add” to create the rule chain.

Open the *Alarms* rule chain. Add a “script” *node* and name it “TempThreshold”. You will notice that this *node* allows you to modify or add JavaScript code to trigger the *alarm* when your data reaches a certain value. The default value triggers the *alarm* when the temperature is greater than 20 degrees, but you can modify this value as needed. Select “Add” to add the TempThreshold *node* to your Rule Engine.

Your rule chain will come with an Input *node* automatically set up. Connect the Input and the TempThreshold *nodes*. This will allow the *Alarms* rule chain to receive input data from your *devices*.



Add a “clear *alarm*” *node* and name it Clear*Alarm*. Modify the default JavaScript code as follows:

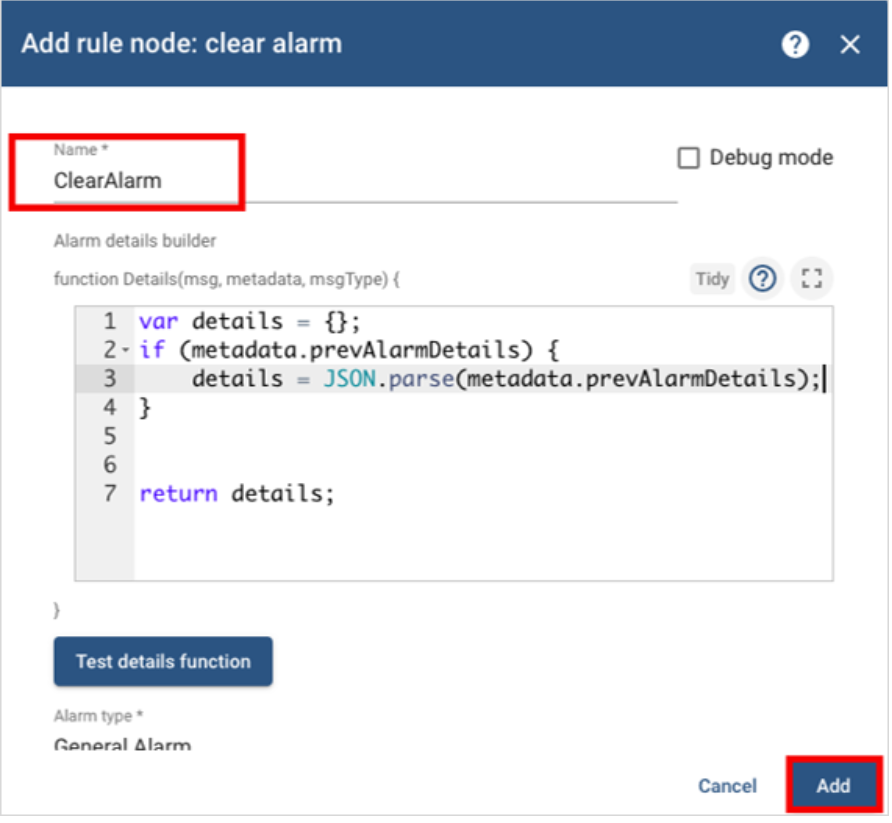
var details = {};

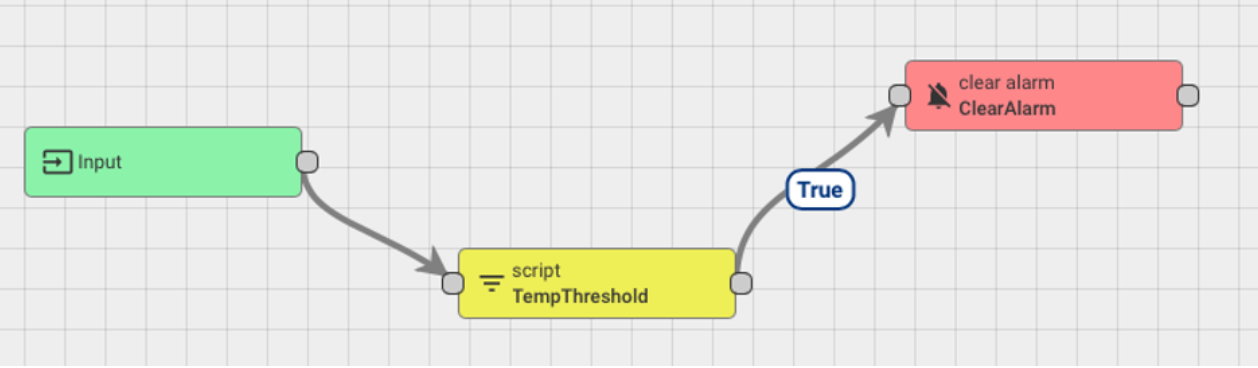
if (metadata.prevAlarmDetails) {

details = JSON.parse(metadata.prevAlarmDetails);}

return details;

Select “Add” to add the Clear*Alarm* *node* to your Rule Engine.



Connect the “TempThreshold” and “Clear*Alarm*” *nodes*, and add “True” as the link label.Add a “create *alarm*” *node* and name it Create*Alarm*. Modify the default JavaScript code as follows:

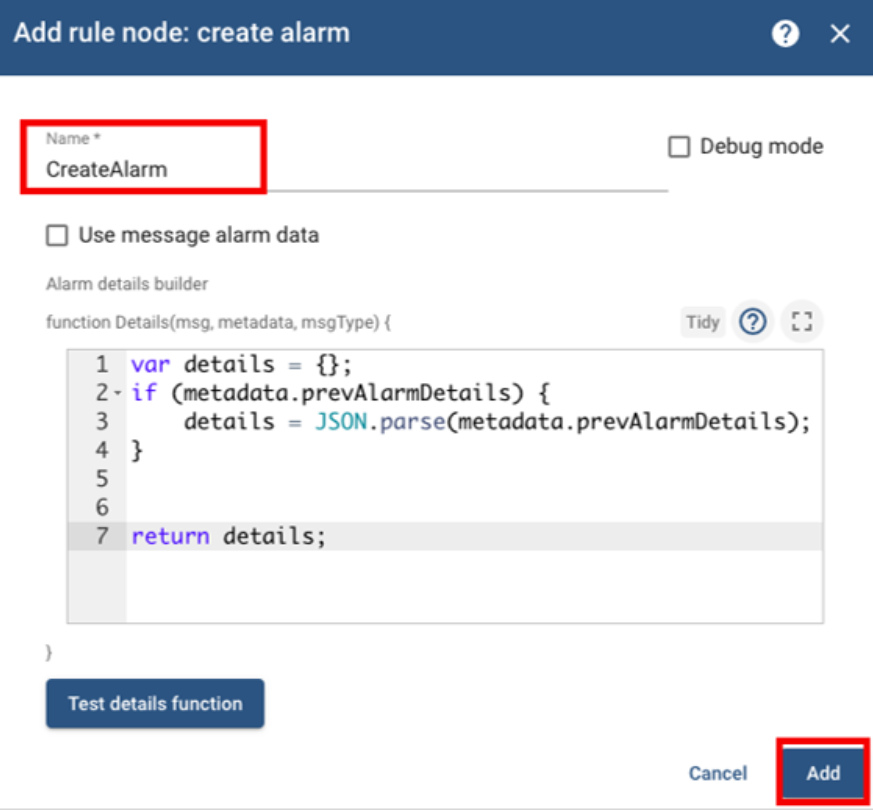
var details = {};

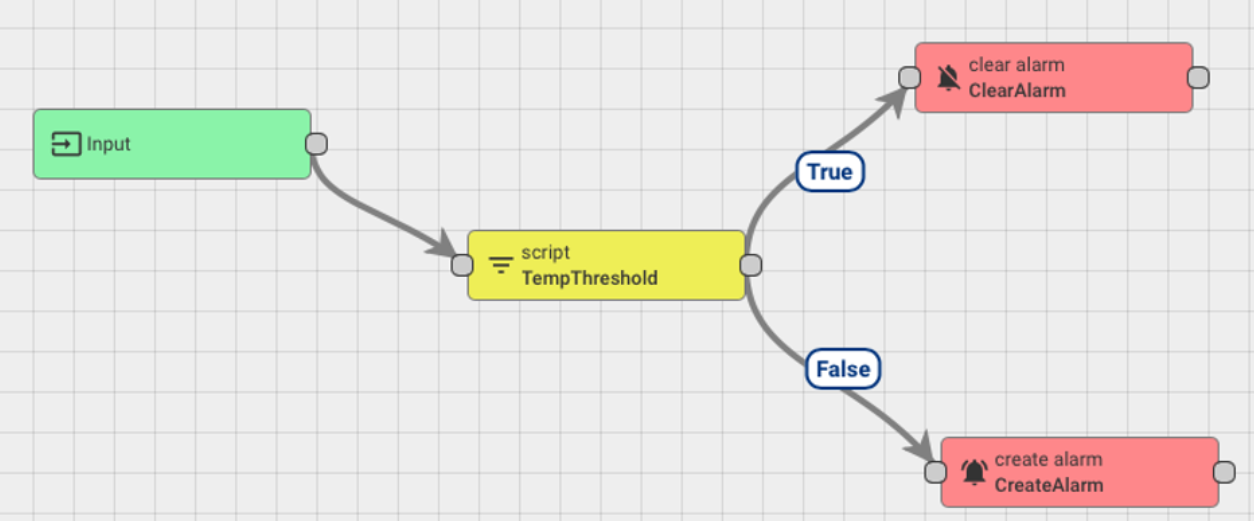
if (metadata.prevAlarmDetails) {

details = JSON.parse(metadata.prevAlarmDetails);}

return details;

Select “Add” to add the Create*Alarm* *node* to your Rule Engine.



Connect the “TempThreshold” and “Create*Alarm*” *nodes,* and add “False” as the link label.

Now that you know how to set up a basic rule chain to create *alarms* in ThingsBoard, you are ready to learn how to connect the *alarm* rule chain to your Root Rule Chain. Then, you can use it to monitor data and send data and *alarm* readings to Firebase.

**Project 24.2 Connections**

In [Project 24.2](https://classroom.emeritus.org/courses/10605/assignments/246368), you will use the *alarm* rule chain to monitor the values of your incoming *stream* of live data and trigger an *alarm* to your Firebase database whenever the temperature gets above a certain threshold.

More broadly, as a data engineer, understanding how to set up *alarms* and *alarm* rule chains in ThingsBoard will be important for you to monitor your incoming data and potentially troubleshoot any problems you may notice with your data. To understand this better, suppose your *device* is constantly sending critically high values. This may be due to a problem with your code, your ThingsBoard configuration, or the physical device itself. Setting up *alarms* can help mitigate this type of problem.