

Lab Guide

IoT Intelligence and Automation with ServiceNow

Bryan Barnard & Jason McKee

Lab instance (1 per TEAM): <https://clabs.link/iot-hyd>

admin / Password: hyd-cc17

technician / hyd-cc17

[[iot.webservice](#) / hyd-cc17]

This
Page
Intentionally
Left
Blank

Lab Goal

In this lab, you'll learn how to build automation quickly and easily across the people and systems of the vast internet of things by using customizable REST integrations and workflows.

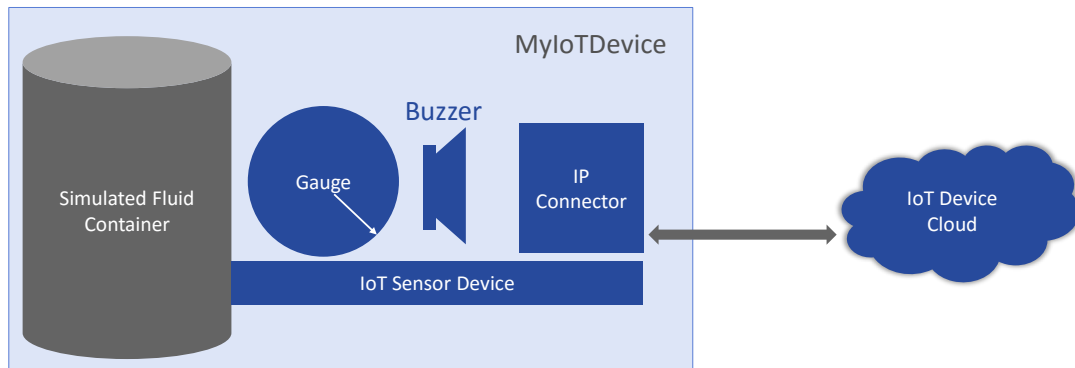
This lab explains the process of automating IoT related incident creation and remediation activities in three steps:

	Incident Creation	Remediation
Lab 1	Manual	Manual
Lab 2	Automatic	Manual
Lab 3	Automatic	Automatic

Lab Overview

IoT devices in this lab measure a fluid level and indicate the current reading through a gauge. In case of a low reading, a sound alarm is generated by the device.

Device Schematics



Once this error state occurs, a team member must physically access the device and go through a procedure to return it to a normal working state. More about the actual procedure is described in the following pages.

The lab uses a team-based approach to go through its steps. Two roles are to be distinguished from each other:

- **Field technician:** can physically observe and access the IoT device
- **Servicedesk:** operates and administers the ServiceNow platform through its web-interface

Lab Goal

This lab simulates that an error is detected at the IoT device level followed by a manual incident creation in ServiceNow.

- The IoT device detects a critical reading and a warning sound occurs, this is reported to the service desk agent
- Lab 1.1 – The service desk agent manually creates an incident in ServiceNow
- The field technician is assigned to resolve the incident
- Lab 1.2 – Instructions for resolution are contained in a knowledge base article
- After successful resolution, the incident is closed and the total resolution time is tracked

Lab 1 Manual Incident Creation and Resolution

Get Familiar with the IoT Device

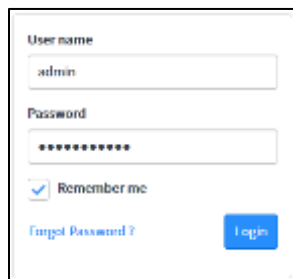
1. Locate the IoT sensor device in the lab room:
 - The visible gauge represents the current reading of a coolant level meter which again is part of an engine.
 - The buzzer sounds only when the current reading falls short of a defined threshold.
2. The button confirms a check-and-refill procedure to remediate error condition.

About the IoT device:

- The device uses an integrated Wi-Fi chip to establish IP-based connectivity.
- In our case, the device is powered by a 5V DC source.
- Several I/O ports are available and can be configured on device level for input vs. output and analog vs. digital. Some ports are PWM-capable. For this lab, I/O ports are pre-configured without the need to be changed.

Lab Instance

1. Navigate to your team's unique lab instance through the provided URL.
2. Log on using the credentials **admin / Knowledge17**



A login form with the following fields and elements:

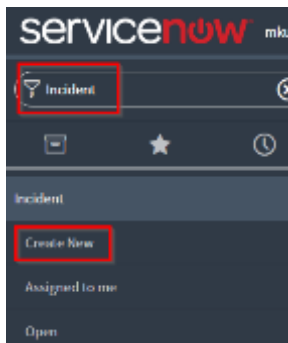
- User name:** A text input field containing the text "admin".
- Password:** A password input field with masked characters (dots).
- ☒ **Remember me**
- [Forgot Password?](#)
- Login** button

Monitor the IoT Device

1. A team member is positioned near the device.
2. Notify the service desk agent manually once the alarm occurs.

Service Desk Agent: Create Incident Manually

1. Navigate to **Incident > Create New**.




- Fill out the relevant details of the incident form:

Caller: **System Administrator** Contact Type: **Walk-in**
 Category: **Inquiry / Help** Assignment Group: **Hardware**
 Subcategory: **IoT Device** Assigned to: **Field Technician**
 IoT Device ID: **Device <team number>**
 Business Service: **Engine**
 Configuration Item: **Coolant Level Meter**
 Short Description: **IoT Device has low coolant level**

The screenshot shows the 'Incident' form in ServiceNow. The following fields are filled out and highlighted with red boxes:

- Number: INC0010029
- Caller: System Administrator
- Category: Inquiry / Help
- Subcategory: IoT Device
- IoT Device ID: Device <rs>
- Business service: Engine
- Configuration item: Coolant Level Meter
- Short description: IoT Device has low coolant level
- Contact type: Walk-in
- State: New
- Impact: 3 - Low
- Urgency: 3 - Low
- Priority: 5 - Planning
- Assignment group: Hardware
- Assigned to: Field Technician

- Save the incident record by selecting **Save** from the  menu:

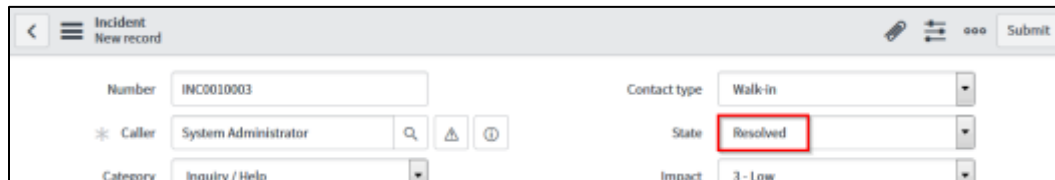


- Scroll to the related search results section of the incident form and locate the Knowledge Base article that appears helpful for the incident's resolution. Click **Preview** to view the content of the article.

The screenshot shows the 'Related Search Results' section. A Knowledge Base article is listed with the title 'Procedure to refill and re-set IoT device...'. The 'Preview' button is highlighted with a red box.

- Service Desk Agent provides details to Field technician, who performs the steps on the device needed to remediate the issue.

- When the device indicates a normal state, return to the Service Desk Agent to let them know they can mark the Incident as **Resolved**.




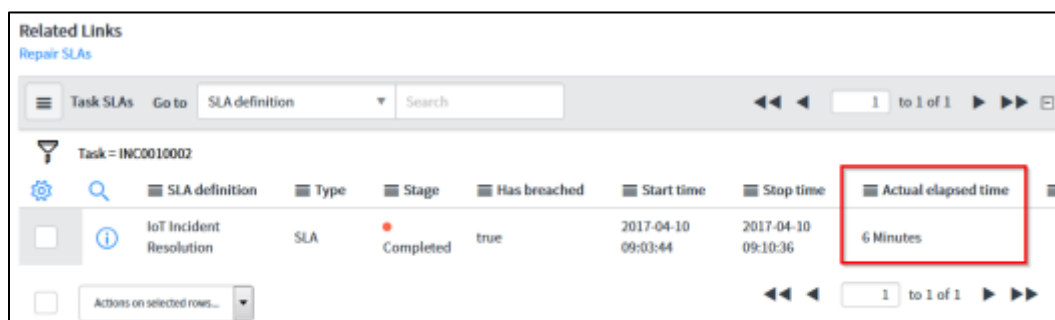
- By default in ServiceNow, the closure of a ticket requires close notes to be added for future reference. Fill in appropriate notes in the Closure Information section:

Close code: **Solved (Permanently)**

Close notes: **Coolant refilled**



- Save the incident record by selecting **Save** from the  menu.
- Scroll down the Incident form and look at the SLA section. The SLA clock starts with the opening of the incident and stops when the incident is marked as resolved. The total processing time is recorded in the right column.



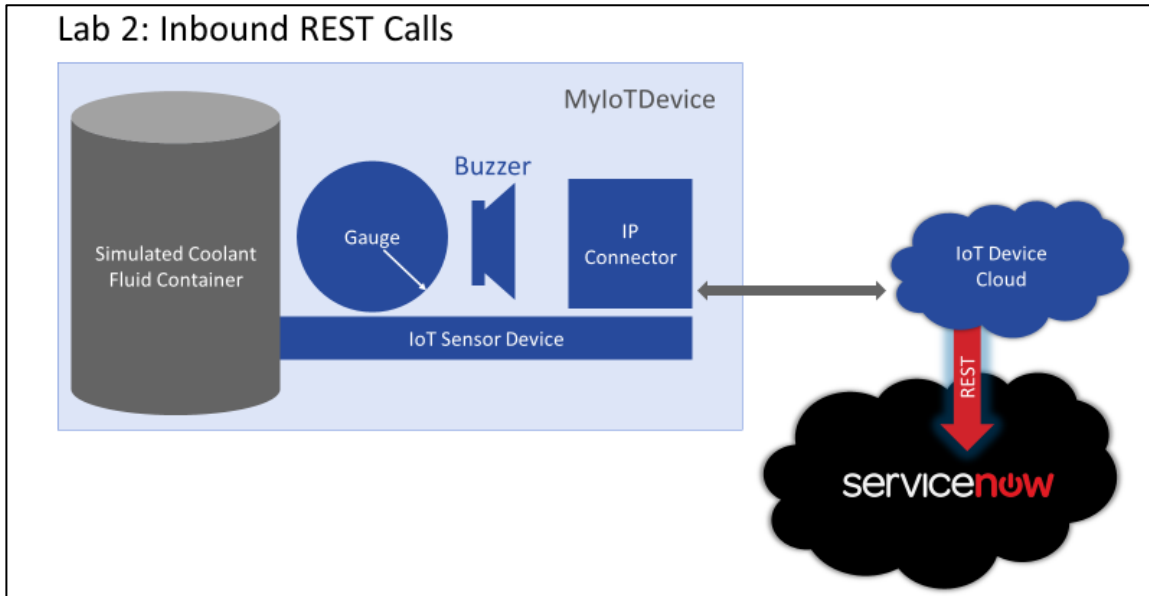
Task	SLA definition	Type	Stage	Has breached	Start time	Stop time	Actual elapsed time
INC0010002	IoT Incident Resolution	SLA	Completed	true	2017-04-10 09:03:44	2017-04-10 09:10:36	6 Minutes

Lab Goal

As before, the device detects a critically low reading. In this lab, if a critical reading occurs, it is reported into the ServiceNow lab instance through a REST API call. Based on that, an incident is created through a business rule.

The process leverages the ServiceNow Table API, a web service that is available on every instance. Once appropriate credentials have been explicitly configured, the Table API allows external sources to gain access on a table specified through the URL.

Lab 2 Automated Incident Creation and Manual Resolution



- Lab 2.1 – Get familiar with the table structure where device critical readings are posted
- Lab 2.2 – Activate a business rule to create an incident when the threshold breach occurs
 - The field technician is assigned to resolve the incident
 - Look at the business service map
 - After successful resolution, the incident is closed and the total resolution time is tracked

Lab Goal

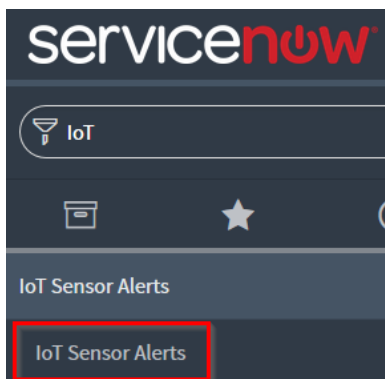
Get Familiar with the Table That Contains Sensor Data

With the web service user access established, the lab instance can receive data from the IoT device. Once the sensor detects a critical level, a record is pushed into a custom table of the ServiceNow lab instance:

u_iot_sensor_alerts

Lab 2.1 Sensor Readings Table

1. Navigate to the table **IoT Sensor Alerts > IoT Sensor Alerts**.



2. Look at the readings that have already been transmitted. Please note, that for this lab only critical readings are reported from the device to ServiceNow.

A screenshot of the ServiceNow web interface showing the 'IoT Sensor Alerts' table. The table has columns for 'Device ID', 'Message', 'Sensor', and 'Sensor Reading'. The first row shows a device ID 'K17_G001', a message 'Oil Temp has reached critical threshold', a sensor 'Oil Temperature', and a reading '341'. The table is currently showing 1 of 1 records.

Device ID	Message	Sensor	Sensor Reading
K17_G001	Oil Temp has reached critical threshold	Oil Temperature	341

Although the lab devices only transmit Coolant level readings, the table setup allows for a device to report data from different sensors like oil temperature, pressure, etc.

Lab Goal

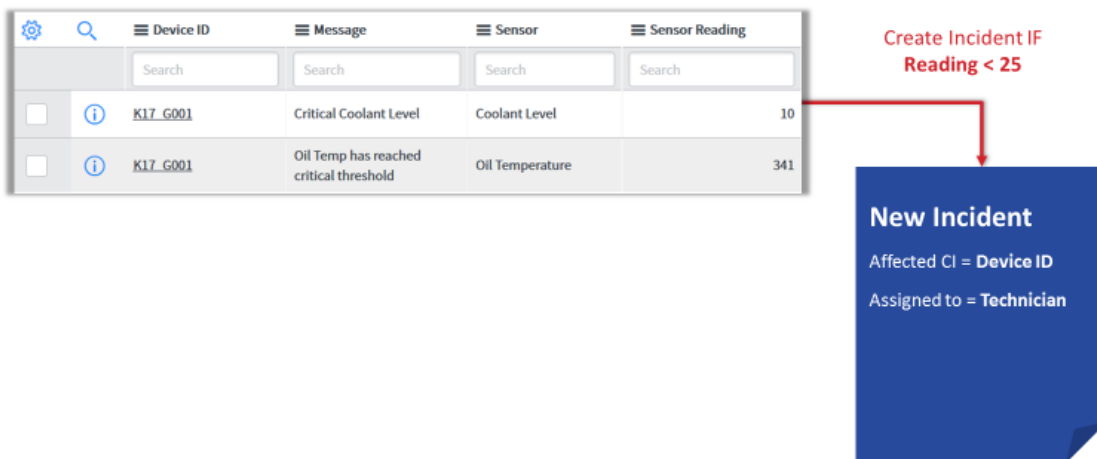
This lab explains how to activate automatic incident creation.

Automatic Incident Creation

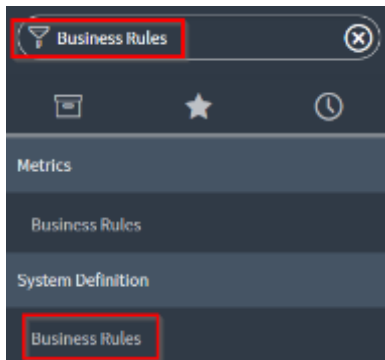
In Lab 1, an incident had to be reported and entered manually. This lab explains how to activate a business rule that creates an incident upon any critical reading reported into ServiceNow.



Lab 2.2 Activate Automatic Incident creation






IoT Incident Creation by a Business Rule



1. Navigate to **System Definition > Business Rules**.



2. Locate the business rule named **Open IoT Incident** by clicking the magnifier icon  and entering the business rule's name. Set the field **active** from false to **true** and confirm the change by clicking the  icon:

	Name	Active	Table	Application
	<input type="text" value="*Open IoT Incident"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>
	Open IoT Incident	<div> <div>false</div> <div>true</div> </div>	 	Sensor Alerts [u_iot_sensor_alerts]
	load globals	false	Business Rule [sys_script]	Global

3. After the activation of the business rule, wait for the device to report a critical reading. The critical state is also indicated by an alarm sound.

Analyze and Remediate Incident

An alarm sound occurs when the device detects the breach of the threshold. The condition is reported into the ServiceNow instance and the business rule is triggered, leading to the creation of an incident.


1. Monitor the list **Incident > Open**. If necessary, refresh the page through the browser.

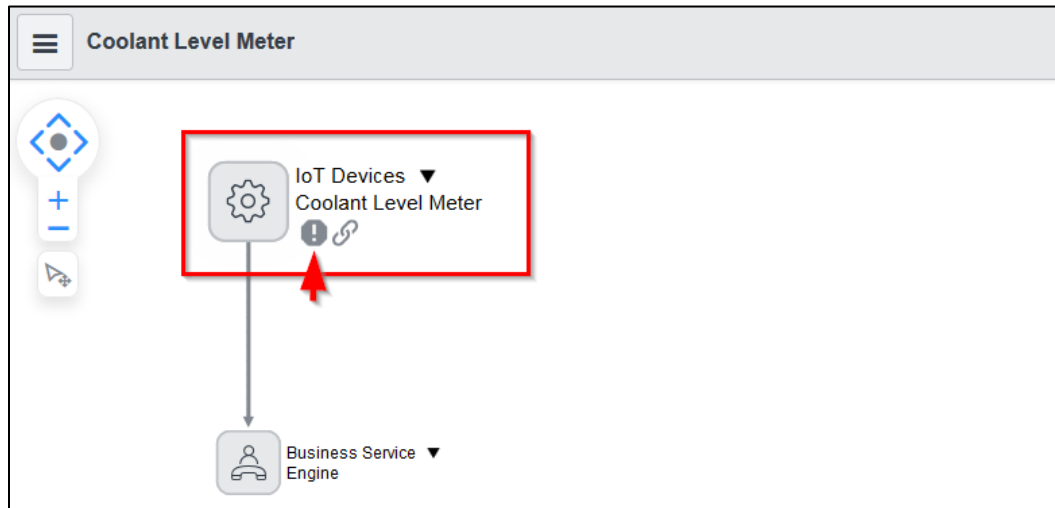
Number	Opened	Short description	Caller	Priority	State	Category	Assignment group
<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>	<input type="text" value="Search"/>
INC0010008	2017-04-11 12:23:30	Auto created: IoT device has low coolant level	System Administrator	5 - Planning		Inquiry / Help	Hardware

An incident tagged **Auto created: IoT device has low coolant level** is added.

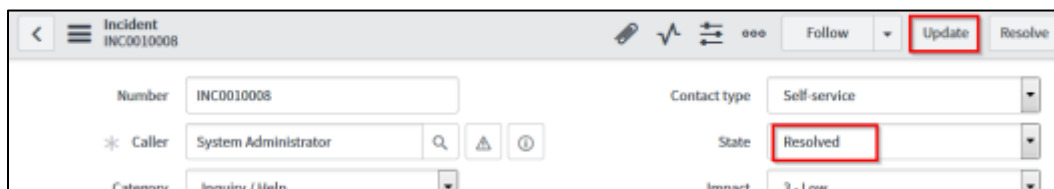
2. Open the Incident by clicking on its number **INCxxxx** in the list view. In the incident form, the fields are populated automatically as far as information is available to the system.

Number	INC0010008	Contact type	Self-service
Caller	System Administrator	State	New
Category	Inquiry / Help	Impact	3 - Low
Subcategory	IoT Device	Urgency	3 - Low
Business service	Engine	Priority	5 - Planning
Configuration item	Coolant Level Meter	Assignment group	Hardware
		Assigned to	Field Technician

- Click the  icon next to the Configuration item field. A new browser tab opens showing the business service and its dependencies on configuration items. In this case, the business service consists of the meter only. Note that mouse-hovering over the exclamation mark brings up additional details about incidents that are opened on this CI.



- Go through the remediation process using the same methodology as in Lab 1 ([goto Lab1 resolution](#)):
 - Locate the device physically
 - Initiate the remediation procedure (refer to KB article if necessary) and wait until the device is back in normal operating state
 - Set the incident state to **Resolved**, add close notes and **Update** the incident record



The screenshot shows the ServiceNow Incident form for incident INC0010008. The form includes fields for Number, Caller, Category, Contact type, State, and Impact. The "State" field is set to "Resolved" and is highlighted with a red rectangle. The "Update" button in the top right corner is also highlighted with a red rectangle. The form also includes a "Follow" button and a "Resolve" button.

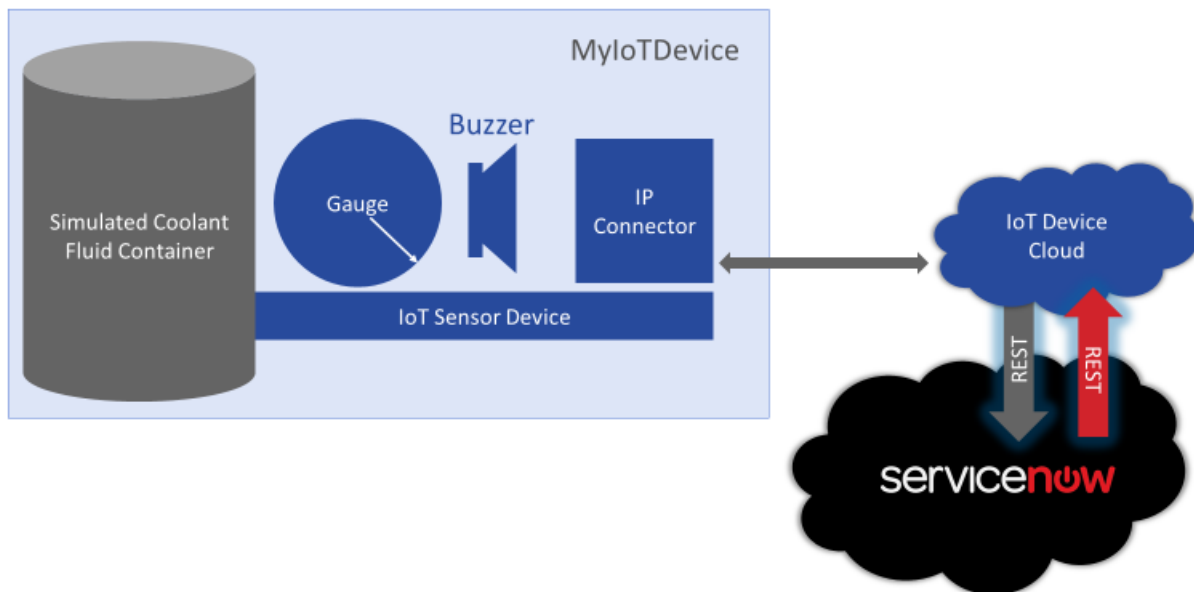
Lab Goal

In this lab, the need to physically walk to the device and trigger the remediation procedure is replaced by an API call from ServiceNow to the IoT device.

The outbound message is sent by the ServiceNow lab instance towards the IoT device cloud, where it triggers the refill process for the device.

Lab 3 Automatic Resolution

Lab 3: Outbound REST Calls



- Lab 3.1 – Take a look at the outbound REST-Message in ServiceNow
- Lab 3.2 – Set up a workflow based on automatically created Incident to trigger the outbound call if the incident qualifies

Lab Goal

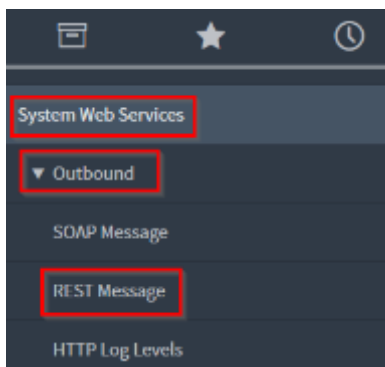
This lab explains the elements of an outbound REST-Message.

Outbound REST Message

An outbound REST message is already defined in the lab instance. For this lab, **thinger.io** is used as intermediary between ServiceNow and the IoT devices. Some elements of the REST message are specific to thinger.io but can be adjusted to other solutions and scenarios. Follow these steps to verify the setup of the REST message.

Lab 3.1 Outbound REST Message

1. Navigate to **System Web Services > Outbound > REST Message**.



2. Click the record **K17 IoT** to open a predefined rest message.

<input type="checkbox"/>	Google Cloud Messaging Send	https://android.googleapis.com/gcm/send	Global
<input type="checkbox"/>	K17 IoT	https://ec2-52-89-162-57.us-west-2.compu...	Global

3. The REST message consists of the following elements:

Endpoint: The URL the message is sent to. Note that it includes a variable device-ID and sensor-ID which need to be replaced on each call of the message:

* Endpoint	https://api.thinger.io/v2/users/..../devices
------------	---

HTTP Request: The elements that get included into the HTTP request header. In the lab setup, an authorization token (**Bearer**) needs to be sent as part of the HTTP header.

Authentication HTTP Request		
HTTP Headers		
	Name	Value
✕ ⓘ	Authorization	Bearer ey.JhbGciOiJIUzI1NiIsInR5cCI6IkpXV...
✕ ⓘ	Content-Type	application/json; charset=UTF-8

HTTP Methods: While two methods are defined, for this lab only the POST method is used to initiate an action on the device.

HTTP Methods			
REST Message - K17 IoT			
	Name	HTTP method	Endpoint
☐ ⓘ	Read Sensor	GET	
☐ ⓘ	Trigger refill	POST	https://api.thinger.io/v2/users/ppemiant...

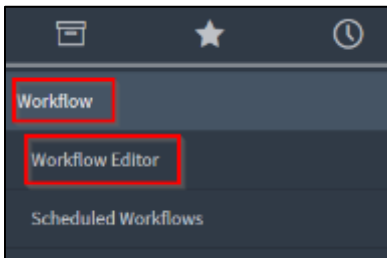
Lab Goal

This lab explains how to create a workflow that leverages the Outbound REST message as defined in Lab 3.1. A call of that REST message simulates the remediation for this lab. The goal is to trigger the remediation process, when an Incident is opened for an IoT Device.

Lab 3.2 Workflow to Trigger Outbound REST-Message

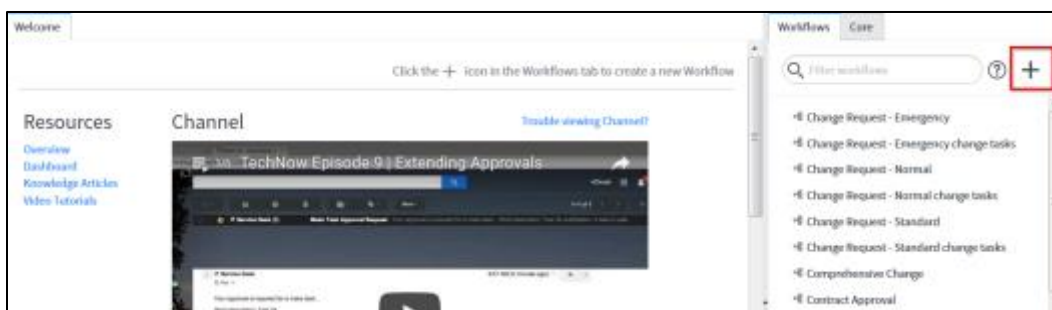
Workflow Editor

1. Navigate to **Workflow > Workflow Editor**.



Note: The Workflow Editor launches in a new browser tab / window.

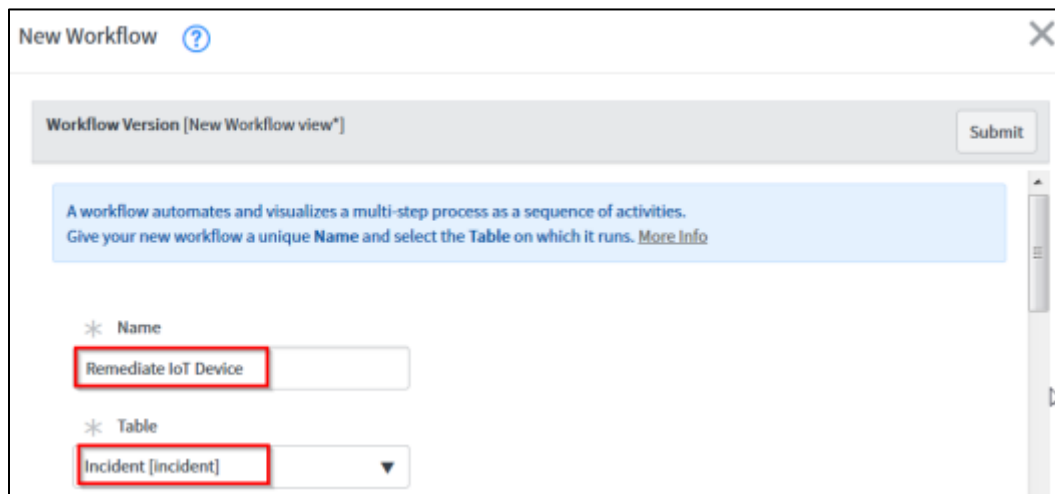
2. Initiate the creation of a new workflow by clicking the **+** icon to the right.



- Enter workflow-specific information into the form:

Name: **Remediate IoT Device**

Table: **Incident [incident]**



New Workflow ?

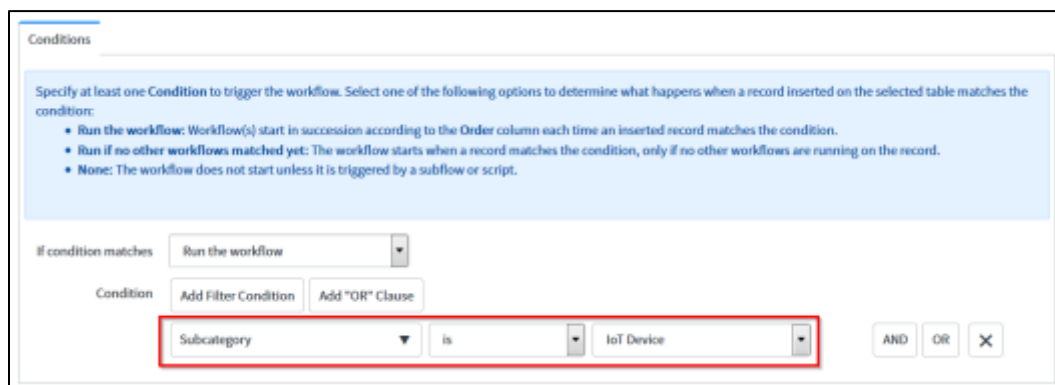
Workflow Version [New Workflow view*] Submit

A workflow automates and visualizes a multi-step process as a sequence of activities. Give your new workflow a unique Name and select the Table on which it runs. [More Info](#)

* Name
Remediate IoT Device

* Table
Incident [incident]

- Scroll to the **Conditions** section and set the following condition:
Subcategory – is – IoT Device



Conditions

Specify at least one Condition to trigger the workflow. Select one of the following options to determine what happens when a record inserted on the selected table matches the condition:

- Run the workflow: Workflow(s) start in succession according to the Order column each time an inserted record matches the condition.
- Run if no other workflows matched yet: The workflow starts when a record matches the condition, only if no other workflows are running on the record.
- None: The workflow does not start unless it is triggered by a subflow or script.

If condition matches Run the workflow

Condition Add Filter Condition Add "OR" Clause

Subcategory is IoT Device AND OR X

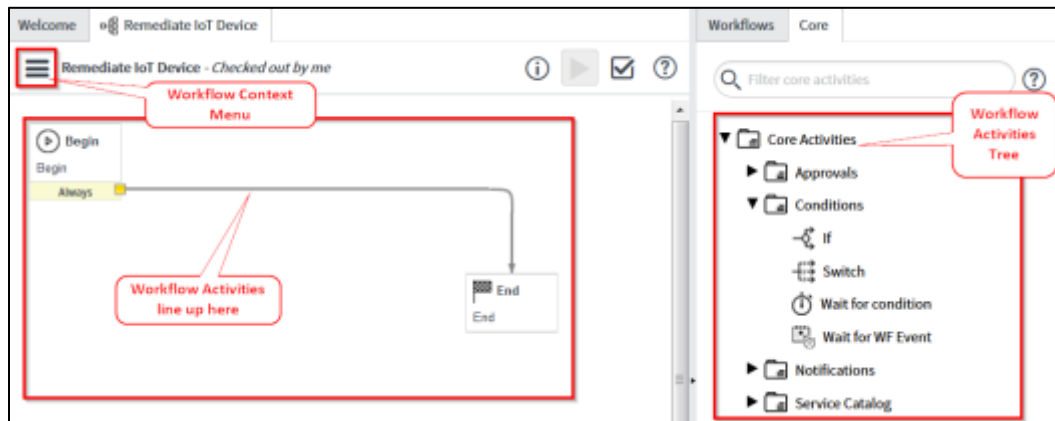
- Click **Submit**.




New Workflow ?

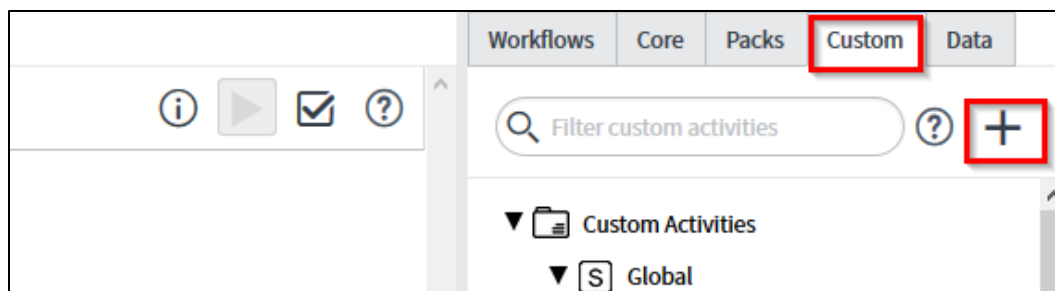
Workflow Version [New Workflow view*] Submit

- After submission, the drawing canvas appears with the following elements:

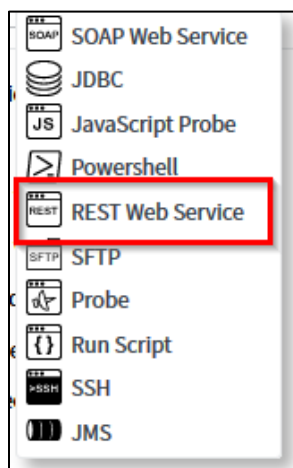


Note: Workflow Elements can be moved into and around the storyboard by drag and drop.

- Locate the **Custom** tab and click the  icon to create a new custom activity from a template.




- Select **REST Web Service** from the pop-up menu.



9. Add name **Trigger Remediation** for the activity and click **Continue**.

The screenshot shows the 'Activity Designer' interface for an activity named 'Trigger Remediation'. The 'General' tab is active, showing fields for Name, Short description, Image, Description, Execution template, Application, Accessible from, and Category. The 'Name' field is highlighted with a red box and contains the text 'Trigger Remediation'. The 'Continue' button at the bottom right is also highlighted with a red box.

10. On the **Inputs** form, click the  icon to add an input variable and specify the fields.

Name: **deviceId**

Type: **String**

Mandatory: **No**

Default: **<<blank>>**

The screenshot shows the 'Inputs' tab of the 'Activity Designer' for 'Trigger Remediation'. It displays a table with columns: Name, Type, Mandatory, and Default. A new input variable has been added with the name 'ABC deviceId', type 'String', and mandatory status 'No'. The 'Continue' button at the bottom right is highlighted with a red box.

Name	Type	Mandatory	Default
{ } Input			
ABC deviceId	String	No	

11. Click **Continue** to proceed to the next section.

12. In the Execution Command section, select the following REST Message and REST Message Function.

REST Message: **K17 IoT**

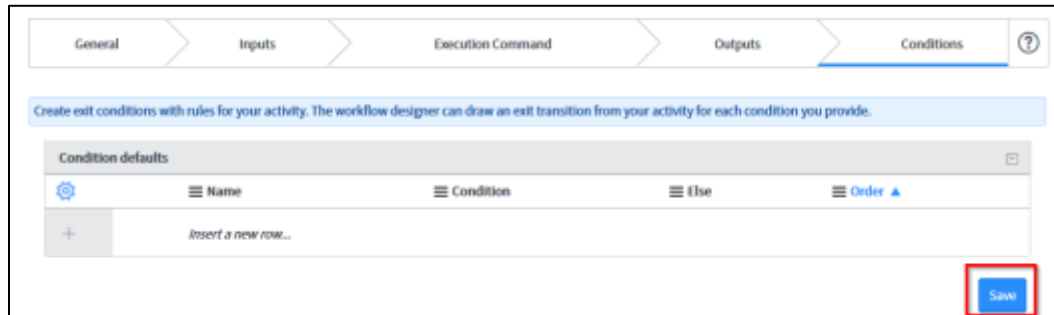
REST Message Function: **Trigger refill**

In the **Variable Substitutions** section drag and drop the deviceID variable into **Value** field [3].

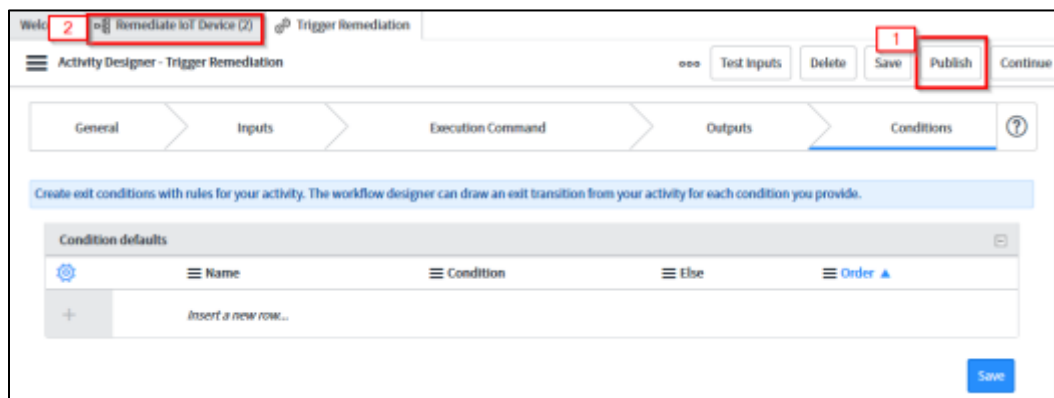
13. Click **Continue** to proceed to the next section.

14. No changes required in the Outputs section. Click **Continue**.

15. No changes required in the Conditions section. Click **Save**.



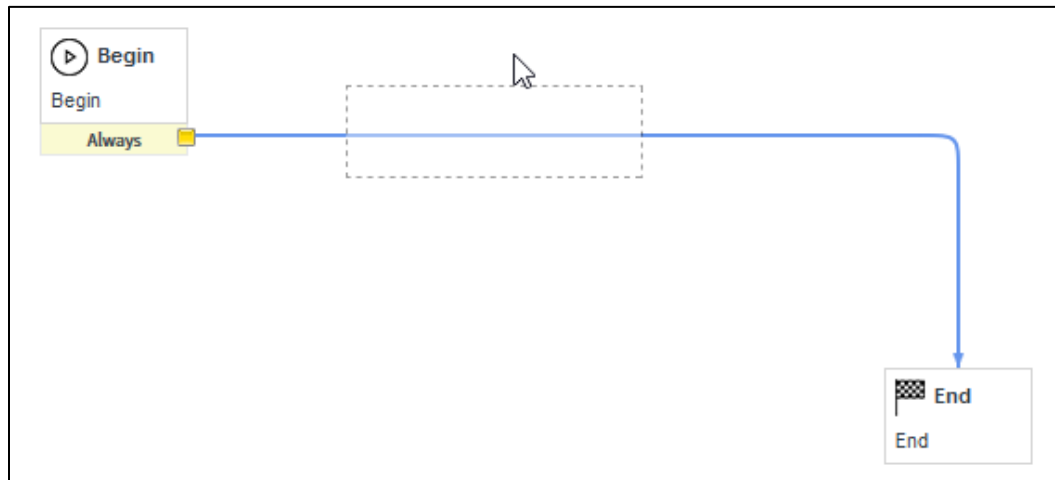
16. Click **Publish** [1] to make the activity available as a workflow action and switch back to the workflow canvas tab **Remediate IoT Device** [2].



17. Find the **Trigger Remediation** action by entering its name in the search field under **Custom** tab. Drag and drop the action into the workflow canvas and snap it in between the start and end position.



18. Note that the workflow activity line switches to a blue color once the activity is positioned correctly to snap in.



19. Fill out the workflow activity details:

Name: **Trigger IoT Device Remediation**

Stage: <<blank>>

Deviceid: `${current. u_iot_device_id}`

And click **Submit**

Activity Properties: Trigger Remediation

Workflow Activity
Trigger IoT Device Remediation [Diagrammer view]

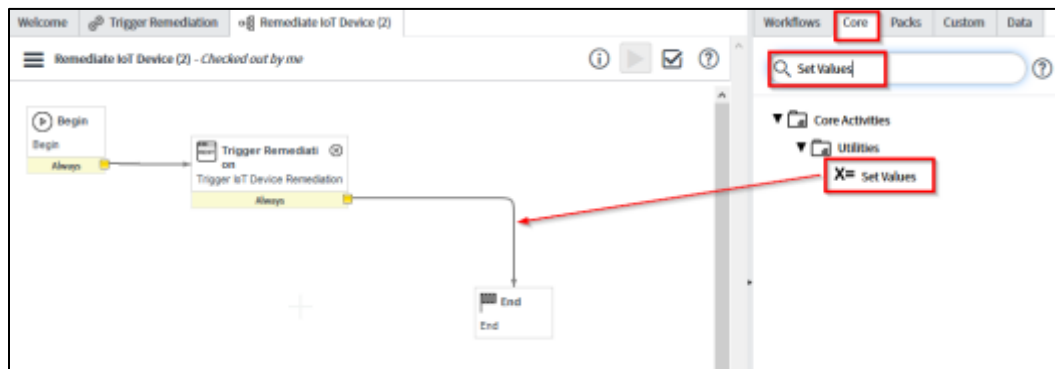
Name: Trigger IoT Device Remediation

Stage:

Deviceid: \$current. u_iot_device_id

20. The Device ID is pulled from the current (incident) record's field named `u_iot_device_id`.

21. The second step in the workflow is to set the incident to a resolved state, after the REST message has been submitted to the device. Find **Set Values** activity from the **Core** tab and pull it into the drawing canvas.



22. In the activity details window, set the following properties:

Name: **Set INT State to Resolved**

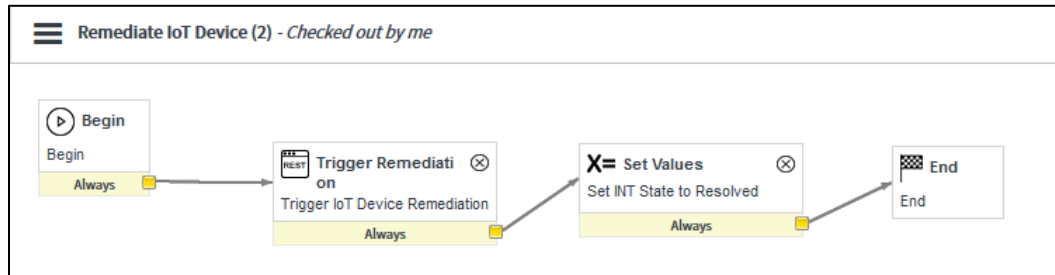
Stage: <<blank>>


Set these Values: **Incident State → Resolved**

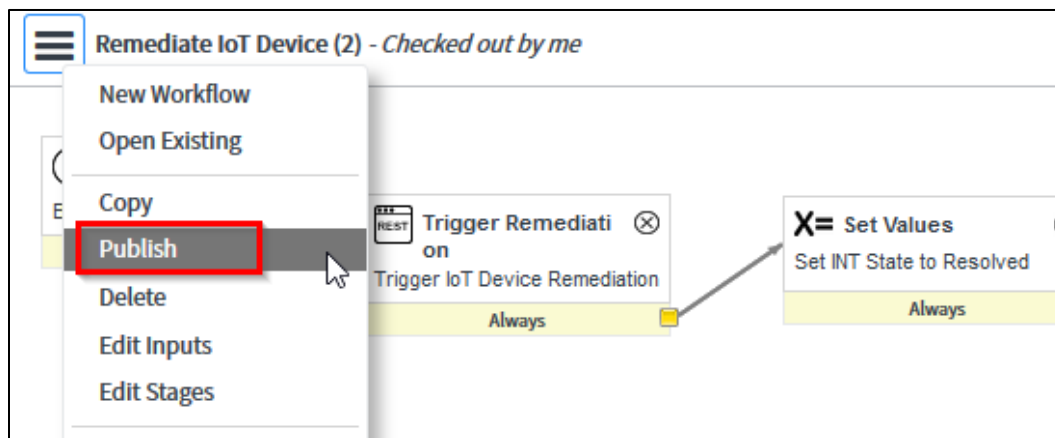
The screenshot shows the 'Workflow Activity' details window for 'Set INT State to Resolved [Diagrammer view]'. The 'Name' field is set to 'Set INT State to Resolved'. The 'Stage' field is empty. The 'Values' section has a description: 'The Set Values activity sets the value of the fields specified below into the current record. [More Info](#)'. Under 'Set these values', there is a dropdown menu set to 'State' and a value dropdown set to 'Resolved'. Below these are two more dropdowns: '-- choose field --' and '-- value --'.

23. Click **Submit**.

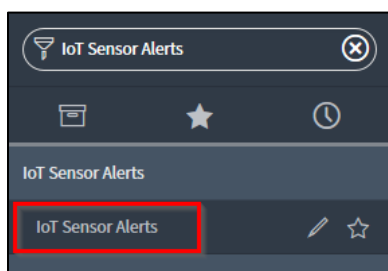
24. The resulting workflow should look like this example.



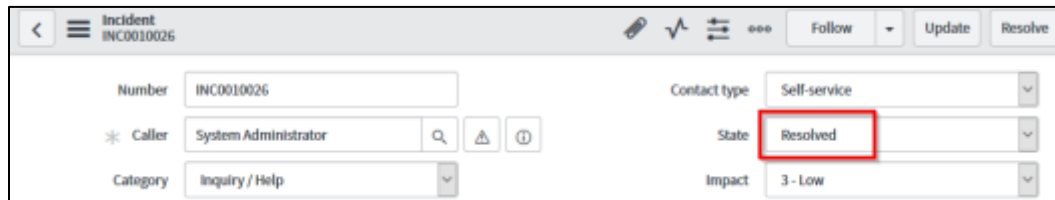
25. To publish the workflow, open the  menu and click **Publish**. Once published the workflow runs whenever an incident is created matching the set criteria.



3. Monitor the IoT Sensor Alerts table.



26. When a new alert is received, it triggers a new incident. The workflow created in this chapter calls the procedure to remediate (refill) the device and sets the incident's **State** to **Resolved** afterwards.



Incident INC0010026		Follow	Update	Resolve
Number	INC0010026	Contact type	Self-service	
* Caller	System Administrator	State	Resolved	
Category	Inquiry/Help	Impact	3 - Low	

Conclusion

In the real world, capabilities in automatic remediation obviously vary depending on the ability to remotely perform the remediation process. This is applicable to such things as a device reset and a firmware upload.

In other scenarios, a physical interaction still is required. The refill of the fluid container, or other mechanical maintenance fall into this category.

For such scenarios, note that although in this lab, device errors raise Incidents, other areas of ServiceNow functionality can be linked. The detection of a critically low fluid level could, for example:

- Raise a case in Field Service Management
- Geo-locate the device
- Assign the refilling task to the preferred vendor that covers the geo-location
- Provide double check capabilities throughout an invoice audit with that vendor