Front-end Challenge

Nested Forms

If coding in Angular 2, use Reactive Forms (Form Builder API) preferably than Template Driven Forms (ngModel)

Create a form that can manage an attributes list and displays the output in JSON format live.

The form will have a fake save button that will get disabled if:

- One of the attributes name is duplicated among them.
- One of the attribute component is invalid (more info below)

Add tabs to the form and distribute the attributes in a given number of categories
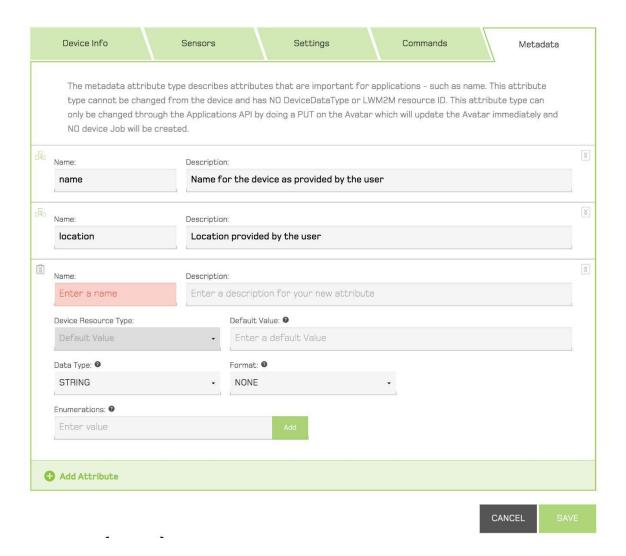
- Categories are static and only to allow the user to easily recognize the attributes.

Internally all the attributes belong to the same list but each one should have a property that allows the tab to render attributes, which belong to the tab's category.
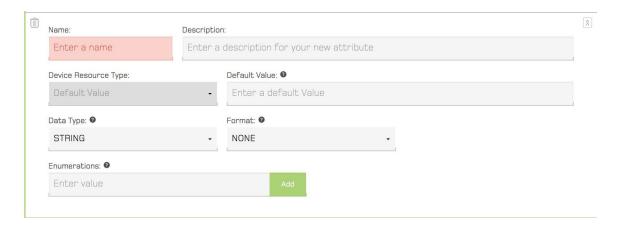
Have an add attribute button to dynamically add attributes in the current selected category.

Each attribute can be removed with a delete icon within the component.

The overall design is not required, use a front end framework to do it (preferably material design)

Device Info	Sensors	Settings	Commands	Metadata

The metadata attribute type describes attributes that are important for applications - such as name. This attribute type cannot be changed from the device and has NO DeviceDataType or LWM2M resource ID. This attribute type can only be changed through the Applications API by doing a PUT on the Avatar which will update the Avatar immediately and NO device Job will be created.

Name:
name

Description:
Name for the device as provided by the user

Name:
location

Description:
Location provided by the user

Name:
Enter a name

Description:
Enter a description for your new attribute

Device Resource Type:
Default Value

Default Value: ?
Enter a default Value

Data Type: ?
STRING

Format: ?
NONE

Enumerations: ?
Enter value   Add

⊕ **Add Attribute**

CANCEL   SAVE

Create an "attribute input" component (custom input component/directive) to encapsulate attribute's validation and easily control the form's save button when an attribute is invalid.
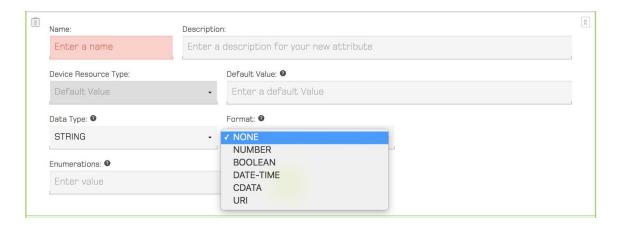
Name:
Enter a name

Description:
Enter a description for your new attribute

Device Resource Type:
Default Value

Default Value: ?
Enter a default Value

Data Type: ?
STRING

Format: ?
NONE

Enumerations: ?
Enter value   Add
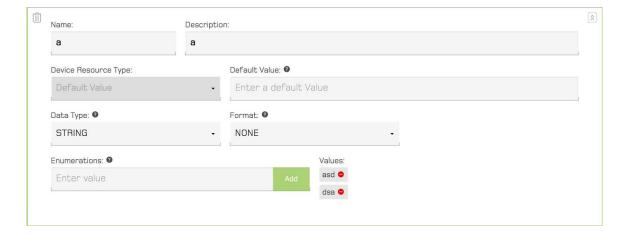
The attribute has all the following properties:

- Name: is required and has to be Unique across all the attributes
- Description
- Device Resource Type: always disabled on this exercise.
- Default Value
- Data Type
- Format
- The rest continues depending on format/data-type values.

Data Type: String

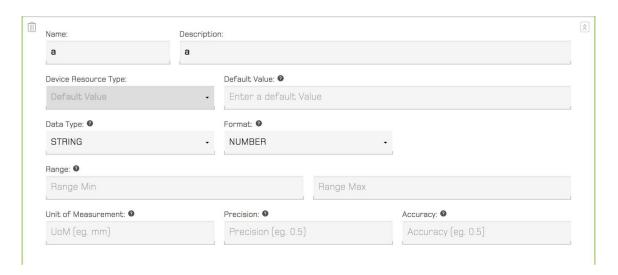- Formats: None, Number, Boolean, Date-Time, CDATA and URI

When "none" is selected it activates an enumerated list of strings:

- Have the enumerations input able to add values to the attribute
- Prevent empty enumerated values from being added to the list

When "number" is selected it activates a ranged set of values:

- Range min/max has to be valid (min cant be gt max, max can be lt min) and these can be either float or integer values. Both are required.
- Precision is the step factor between the defined ranges; its value has to allow the user to go through min to max and not exceed it. Required.
- Accuracy acts the same than Precision.

When "Boolean", "Data-type", "URL" there's no custom behavior:

Data Type: Object

- Default Value is disabled
- Format is disabled.

The attribute object has to always reflect what is displayed in a given combination (do not keep previous combination values in the object, make them null)