# **Product Insert Module**

Assume you are creating an e-Commerce web application, you need to write two different views one for showing options to add different products and one inserting a new entry for specified product.

Every product could be very different from the other, for instance a TV is very different from Shoes and have different attributes. We have pre-created some API endpoints to help you with this exercise.

Your task is following:

**Angular (NOT Angular JS) Web Application:**

1. **Home Screen:** On the home page, call the following endpoint to get the list of the available products

[**https://ds-ecom.azurewebsites.net/products**](https://ds-ecom.azurewebsites.net/products)

Based on the information returned create Add button for each product returned by the above REST call.

For instance: The above call returns the following JSON, so you should create two buttons called “Add TV” and “Add Mobile”.

[

{

"name": "TV",

"definitionUrl": "https://ds-ecom.azurewebsites.net/products/tv"

},

{

"name": "Mobile",

"definitionUrl": "https://ds-ecom.azurewebsites.net/products/mobile"

}

]

1. **Product insert screen:** Based on the button clicked do a REST call to get its definitionUrl and based on the JSON returned create a ***dynamic*** form for each field specified in the JSON.

**For instance,** if this endpoint<https://ds-ecom.azurewebsites.net/products/tv> returns the following JSON, your form should have 3 fields named as Name, Description and IsSmart.

**[**

**{**

**"caption": "Name",**

**"type": "text",**

**"mandatory": true,**

**"defaultValue": "",**

**"validationMessage": "Name is mandatory"**

**},**

**{**

**"caption": "Description",**

**"type": "text",**

**"mandatory": true,**

**"defaultValue": "Please enter something here....",**

**"validationMessage": "Oops! You can't leave it empty"**

**},**

**{**

**"caption": "IsSmart",**

**"type": "bool",**

**"mandatory": true,**

**"defaultValue": "False",**

**"validationMessage": ""**

**}]**

Please see the details below about the field JSON.

1. Caption: this is the name of the field
2. Type: this determines the data type of the field, a text means textbox should be shown in UI, a bool means a checkbox should be shown in UI. You need to at least support these two.
3. Mandatory: This determines whether the field is mandatory or not, if mandatory then validation should be fired with the validation message
4. DefaultValue: Indicates the default value when a new insert view is opened
5. ValidationMessage: The message that we need to show if the field is mandatory

The project should be easy to run with a readme.txt/ readme.md which specifies the steps to run. We prefer you to create a GitHub repository for the answer, alternatively you can zip and send the answers as well.

Following are the evaluation criteria

1. Completeness
2. Correctness
3. Readability
4. Maintainability

All the best!