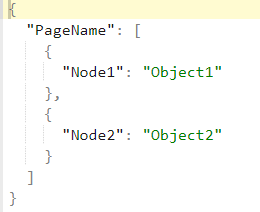
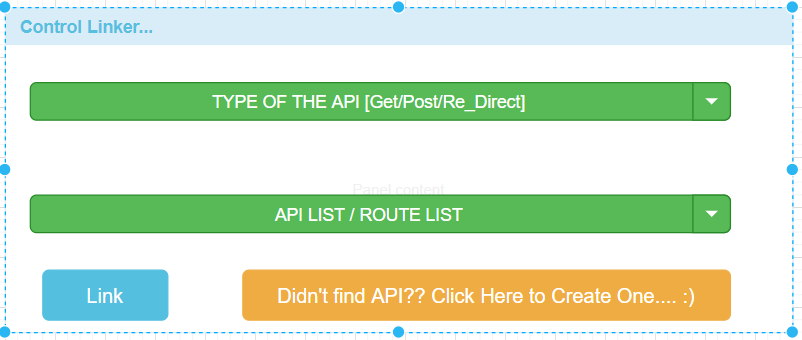
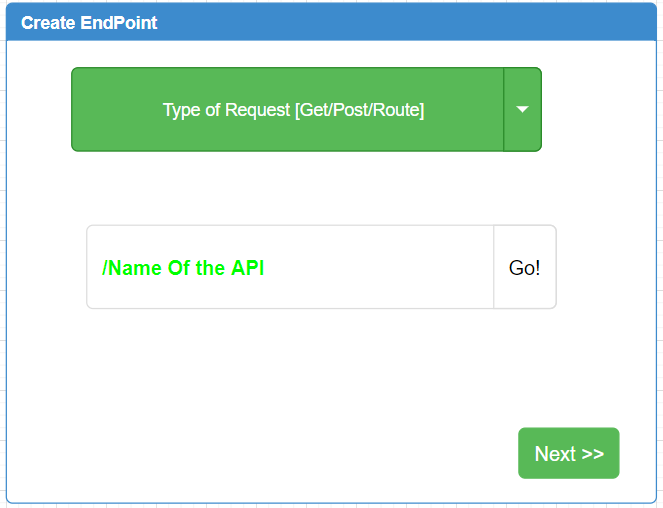


# CREATE UI BLOCK:

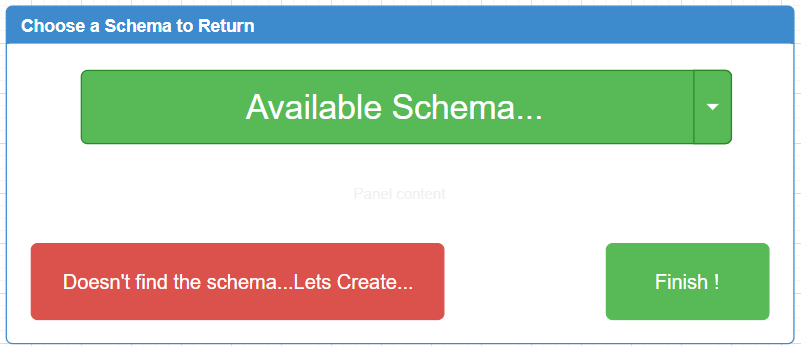
* When user Clicks on the “Create UI Block” button, User should be Prompted to Enter the Name of the UI Block, on Enter the Name of UI Block and Clicked on “OK” button, User should be able to access to a page, which displays only the following
  + Grid System Section
  + Additionally, a FOLDER whose name is the “Name of the UI Block” should be created and a JSON file whose Name is “TreeStructure.json” file should get created inside the “PUBLIC” folder.
* Only when the user chooses an appropriate Grid System, then following Tabs should get displayed
  + Grid System Section [Continue to Display]
  + Form Component
  + Components
  + JavaScript Components
  + Npm Components
  + **Save UI Block - Button**
* A JSON Structure on the **Hierarchy** of the UI Controls should be formed in the BACK END
  + 
* When Dragging and Dropping each of the following component, User should be provisioned an Option to Link the Component with an appropriate API [POST/GET/HARDCODED], in the form of a MODEL POP UP, whose Mock up is provisioned below
* Also, Once the Component Get created, a custom attribute, whose combination is as follows should get generated
  + AutomationId=”RandomNumber”



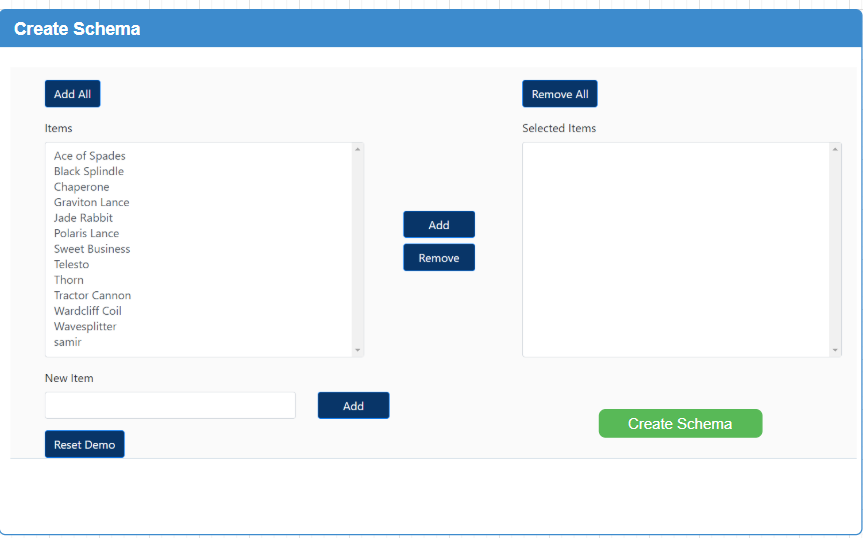
* When User Clicks on the Link Button, the Child Nodes of the JSON Response Object, should be displayed, allowing the User to Choose the Appropriate Child Nodes based on the Component, **For example**, CARD Component will expect 3 Child Nodes, Hence User Should be choosing 3 Child Nodes.
* When user Clicks on “Didn’t find API?? Click Here to Create One…. 😊”, button , user should be prompted with a New Model which is as follows:



* On Clicking on the “**Next >>**” button, User should be prompted with the model, which ask the end user to choose a Schema from the available list



* If User Clicks on the “Finish !” button, the Child Nodes of the JSON Response Object, should be displayed, allowing the User to Choose the Appropriate Child Nodes based on the Component, **For example**, CARD Component will expect 3 Child Nodes, Hence User Should be choosing 3 Child Nodes.
* If the Use Clicks on “Doesn’t find the Schema…Lets Create…” button, then User should be prompted with “**Create Schema**” Model, as displayed below:



* Once User Clicks on the “**Create Schema**” button, the Control should return back to “Choose a Schema to Return” screen, which allows the User to Choose the Newly created schema, and subsequent clicking on “Finish !” button.
* Once User Clicks on “Finish !” button as usual, the Child Nodes of the JSON Response Object, should be displayed, allowing the User to Choose the Appropriate Child Nodes based on the Component, **For example**, CARD Component will expect 3 Child Nodes, Hence User Should be choosing 3 Child Nodes.

# CREATE A PAGE:

* When User Clicks on “CREATE A PAGE” Button, The App should prompt the User to Enter the Name of the Page, Once the User Enter the Name of the Page, a FOLDER of the Same Name should get created.
* Secondly the App should redirect the User to a Page, which has the following
  + Grid System Section
  + Save Page [Button]
* Once the User Chooses the appropriate Grid System, then “**Available Set of UI Blocks**” should get Displayed, below the Grid System Section
* User Should be able to Drag and drop the Created UI BLOCKS in the Appropriate Cols/Row of the Grid System.
* The Json Nodes of Each of The UI Blocks Should get appended in the Order of Drag and Drop
* Save Page

**IDEA ALERT:**

* UI + BackEnd Testing at a Single Time
* **Example**: Lets say that an UI block is displaying a HighChart. now validating a high chart is not possible in UI automation, now, here, we can leverage, a combination of UI + Backend Testing as follows
  + Until Navigating to Highchart, we can use UI automation script, after that validating the HIGHCHART alone, we can make use of Highchart specific API Testing.

# DESIGN DATABASE:

<https://github.com/mrvautin/adminMongo>

Or

<https://webapplog.com/mongoui/>