Screen Object Description:

This link is the entire json collection of the screen.

https://drive.google.com/file/d/1pKPi4GTiH907X2V5uMy3ahE7edJ05Mrm/view?usp=sharing

GridObject:

```
[{
  "grid_fields": {
    "entityId": "",
    "default_field": [
      []
    ],
    "htmlId": "",
    "componentId": "",
    "custom_field": [],
    "event": ""
  },
  "is_grid_present": true
}]
```

1. The grid-fields are used to capture the information for the ag-grid table.
2. The entityId field stores the entityId which is mapped to the grid.
3. The default field captures the default column name of the grid and
4. The htmlId are the ID which are given by grapesjs
5. The componentId is the ID which is given by grapesjs.

6. The custom fields array is used to store the object column name ,column id,entityname and fieldname of the mapped entity to this column.
7. The event field is to capture whether or not we need to load this table like (Onload, AfterLoad).
8. The is_grid_present field is what says whether an ag-grid is added or not .

Gjs - fields:

https://docs.google.com/document/d/1tSkvVOwlGCYodGgWUejZ5ewJmhRPjdkmGyw2GuWQrbQ/edit?usp=sharing

The above link has the entire collection of the gjs objects.

- The object/fields which are prefixed with gjs are all the predefined one which are obtained from grapesjs. To understand on them please check this link https://grapesjs.com/docs/modules/Storage.html#store-and-load-templates
- The fields are gjs-assets that are used to capture the images and videos which have been added in the screen designer.
- The fields gjs-styles captures the css properties from the screen designer and they store the formation of an array of objects as in below. Each individual html component will have its own css styles for them.
- The gjs-component,gjs-html,gjs-css,gjs-styles,gjs-assets are the metadata which we need for generation and so we are using them in the screen microservice.

```
{
            "selectors": [],
            "selectorsAdd": "body",
            "style": {
                "margin-top": "0px",
                "margin-right": "0px",
                "margin-bottom": "0px",
                "margin-left": "0px"
            }
        },
```

- The gjs-component field stores the values of the html of the screen designer and stores in the array of objects where we capture the tag name, class, type and the attribute id for each of them.

- The below is the link for the explanation for the gjs-component which is the more important object which is used in the generation.

  Note:
  https://docs.google.com/document/d/1pJODRL_7aN2Oko98_F5N1dLKpGFegkace3M_ZFsX9O8/edit?usp=sharing

- The gjs-html field captures the html of the screen designer. **This is not used in the generator anywhere we just capture it.**


```
"gjs-html": "<body id=""wrapper""><form id="template-i58h"
class=""form""><div id="template-i7vg" class=""form-group""><label
id="template-i2v7" class=""label"">TicketName</label><input
placeholder="" id=""template-ihza"" class=""input
form-control""/></div><div id="template-iog1"
class=""form-group""><label id=""template-iyig""
class=""label"">TicketStatus</label><input type=""email""
placeholder="" id=""template-ihnj"" class=""input
form-control""/></div><div id="template-i47fr"
class=""form-group""><button type=""submit"" id="template-iwf2q"
class=""button btn btn-primary"">Update</button><button
id=""template-ifqoo"" class=""button btn
btn-primary"">Delete</button></div></form></body>"
```


- The gjs-css field captures css changes which have been added in for the screen designer like button color and and text width and height etc and it is used in the generator to add the component css.

Component-Lifecycle:

- The component lifecycle is what captures lifecycle part of the page which has been created on the screendesigner based on the UI framework. Like whether to load this page onLoad , onview has initialized.

Project:

- The project is what captures the project id for the screen to know that this screen belongs to this project.

Feature:

- The feature is what captures the feature id for this screen collection.

ScreenType:

- The screen type is what captures whether it is for mobile type or desktop type.

isTemplate:

- The istemplate is a boolean which mentions whether it is a template or a screen.

Stylesheet:

- The stylesheet is an array which captures the stylesheet links which are used for the template design like the bootstrap stylesheet cdn etc currently and we are not using them.

Scripts:

- The script array is what captures the script, the link which is used for the template design currently and we are not using them.

Css-guidelines:

- The css-guidelines is what captures the templates design currently and we are not using them.

Flow_info:

- The flow info is an array which captures details of a flow which is mapped with the html attribute.
- The below is the sample details of the flow infor array which captures the htmlId, componentId which is mapped with the gjs-component.
- The element which we get from the grapesjs metadata based on the block he chose.
- The verb is what mentions what type of action the flow needed.
- The event is what captures on what type of the event they have selected in the grapesjs traits.
- The flowName is what captures what flow they have selected.
- The id is uuid which is used to specify the uniqueness of the flow.
- The flow is what stores the flowId of the selected flow.


```
"flows_info": [
        {
            "htmlId": "template-iwf2q",
            "componentId": "c120323",
            "elementName": "button_template-iwf2q",
            "verb": "click",
            "event": "",
            "flowName": "GpUpdate",
```

```
            "_id": "5f7880ec746fbe1ed1825958",
            "flow": "8d3e0e30-057c-11eb-9685-132994e91120"
        },
        {
            "htmlId": "template-ifqoo",
            "componentId": "c120341",
            "elementName": "button_template-ifqoo",
            "verb": "click",
            "event": "",
            "flowName": "GpDelete",
            "_id": "5f7880ec746fbe1ed1825959",
            "flow": "8d3e8361-057c-11eb-9685-132994e91120"
        },
        {
            "htmlId": "",
            "componentId": "",
            "elementName": "",
            "verb": "",
            "event": "OnLoad",
            "flowName": "GpGetNounById",
            "_id": "5f788111746fbe1ed182595e",
            "flow": "a331a330-7619-11e9-8b28-fb1133df238c"
        },
    ]
```


Entity_info:

- The entity info table captures the entity binding which are done from the HTML attributes.
- The entity info array has a field object which is the entity fieldId and field details from the entity.
- The htmlId,componentId and the elementName are captured from the grapesjs metadata.
- The entityId is the entityId from the entity table and the _id is uuid which we generate for this entity.


```
    "entity_info": [
{
  "fields": {
      "fieldId": "73dae120-057c-11eb-93ba-f7b06ce772d0",
      "name": "Ticketname",
      "description": "Description",
      "typeName": "Text",
```

```
            "dataType": "String"
        },
        "htmlId": "template-ihza",
        "componentId": "c120253",
        "elementName": "input_template-ihza",
        "entityId": "1bfcad80-057c-11eb-93ba-f7b06ce772d0",
        "_id": "5f7880ec746fbe1ed182595a"
    },
    {
        "fields": {
            "fieldId": "73db0830-057c-11eb-93ba-f7b06ce772d0",
            "name": "Ticketstatus",
            "description": "Description",
            "typeName": "Text",
            "dataType": "String"
        },
        "htmlId": "template-ihnj",
        "componentId": "c120297",
        "elementName": "input_template-ihnj",
        "entityId": "1bfcad80-057c-11eb-93ba-f7b06ce772d0",
        "_id": "5f7880ec746fbe1ed182595b"
    }
]
```


Route_info:

- The route_info is what captures which screen you want to route too.
- The htmlId,componentId and elementName are what we get from the grapesjs metadata.
- The screenId captures the screenId of the screen which you want to route.
- The screenName stores the name of the route screen.
- The routeType is stored whether it is query parameter or path parameter.
- The methodname is what stores the route flowname.
- The screenflow is the flowId.
- The screenFlowName is the flow which you want to use for the route screen.


```
[
    {
        "htmlId" : "template-ioeti",
        "componentId" : "c76134",
        "elementName" : "grid_template-ioeti",
        "screenId" : "52eb11a0-0eec-11eb-bd8f-5fec25f3b4bc",
        "screenName" : "screen642369",
        "routeType" : "queryParameter",
        "methodName" : "GpRoute",
```

```
        "screenFlow" : "af257040-0ee8-11eb-aa9c-6902e74148b4",
        "screenFlowName" : "GpUpdate",
        "_id" : ObjectId("5f88573f1a44332dc3c03f85")
    }
]
```