GjsComponent:

The below object is the complete object of gjs-component which is mainly used in the UI generation and this object can be used for all the UI generation.

The gjs-component captures the entire html attributes of the grapes-js screen designer.

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

The gjs-component has an array object in component object which captures the html elements like the tag name, class, type of the html element and gives ID for each of the elements.

And also captures the entity for that html element which the entity is mapped so that when we generate it we will be able to map entity ID from the entity_info object which also stores the same attribute id for the data binding.

The below is the sample of the one of html attributes which captures the input and form fields of the html which will be inside the component array object.

```
{
  "content": "",
  "classes": [
    {
      "name": "form-group",
      "label": "form-group",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    }
  ],
  "attributes": {
    "id": "template-i7vg"
  },
  "components": [
    {
      "type": "label",
      "content": "",
      "classes": [
        {
          "name": "label",
          "label": "label",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    }
  ],
  "attributes": {
    "id": "template-i2v7"
  },
  "components": [
    {
      "tagName": "",
      "type": "textnode",
      "removable": false,
      "draggable": false,
      "highlightable": 0,
      "copyable": false,
      "content": "TicketName",
      "attributes": {
        "id": "template-iaiea"
      },
      "_innertext": false
    }
  ]
},
{
  "type": "input",
  "name": "input_template-ihza",
  "void": true,
  "content": "",
  "classes": [
    {
      "name": "input",
      "label": "input",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    },
    {
      "name": "form-control",
      "label": "form-control",
      "type": 1,
      "active": true,
          "private": false,
          "protected": false
        }
      ],
      "attributes": {
        "placeholder": "",
        "id": "template-ihza"
      },
      "entity": "1bfcad80-057c-11eb-93ba-f7b06ce772d0"
    }
  ],
  "open": 0
}
```

The above object of the gjs-component will be inside the component array as in below.

The first set of objects captures what the html format is like is a form , input etc.

```
{
  "tagName": "form",
  "type": "form",
  "content": "",
  "classes": [
    {
      "name": "form",
      "label": "form",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    }
  ],
  "attributes": {
    "id": "template-i58h"
  }
Component: [
]
}
```

Based on the above gjs-component is mostly used for the generation of the UI in the generator.

The values inside the component is what is used to build the html element and map with the data binding for each of entity fields.

The gjs-components component array is what is used to construct the HTML, and adding class from the properties which have been stored in the class array as in below.

```
  "classes": [
    {
      "name": "input",
      "label": "input",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    },
    {
      "name": "form-control",
      "label": "form-control",
      "type": 1,
      "active": true,
      "private": false,
      "protected": false
    }
  ]
```

The same gjs-component is used to add data binding, mapping flows. The gjs-component data binding and flow mapping are mapped from matching the values from the entity_info object and flow_info object.