
developer.skatelescope.org

Documentation

Release 0.1.0-beta

MAXIV/SKA Organization

Mar 04, 2022

Contents

1	Data model for storing the dashboards	3
2	Data model for storing the user action log details	5

This project provides a central store used by the Taranta suite for storing and sharing the dashboards created by taranta users.

It consists of:

- A data store (implemented as a MongoDB database)
- A set of RESTful services used by the Taranta application to create, retrieve, update and delete individual dashboards associated with a particular user and access to one or more user groups.
- Basic action logging service / store for Taranta actions

The 'models' contains the data structures used to store the two types of information

CHAPTER 1

Data model for storing the dashboards

```
1 let dashboardSchema = mongoose.Schema({
2   _id: {
3     type: ObjectId,
4     auto: true,
5   },
6   name: {
7     type: String,
8     required: true,
9   },
10  user: {
11    type: String,
12    required: true,
13  },
14  widgets: {
15    type: [WIDGET],
16    required: true,
17  },
18  variables: {
19    type: Object,
20    required: false,
21    default: [{}],
22  },
23  insertTime: {
24    type: Date,
25    default: Date.now,
26  },
27  updateTime: {
28    type: Date,
29    required: true,
30    default: Date.now,
31  },
32  group: {
33    type: String,
34    required: false,
```

(continues on next page)

(continued from previous page)

```

35     default: null,
36   },
37   groupWriteAccess: {
38     type: Boolean,
39     required: false,
40     default: false,
41   },
42   lastUpdatedBy: {
43     type: String,
44     required: false,
45     default: null,
46   },
47   deleted: {
48     type: Boolean,
49     default: false,
50   },
51   tangoDB: {
52     type: String,
53     default: '',
54   },
55 });

```

```

1  const WIDGET = {
2    canvas: String,
3    id: String,
4    x: Number,
5    y: Number,
6    height: Number,
7    width: Number,
8    type: { type: String },
9    // Necessary since `type` is a reserved word in Mongoose
10   inputs: Schema.Types.Mixed,
11   order: Number,
12 };

```


CHAPTER 2

Data model for storing the user action log details

```
1 let userActionLog = mongoose.Schema({
2   actionType: {
3     type: String,
4     required: true,
5   },
6   timestamp: {
7     type: Date,
8     required: true,
9     default: Date.now,
10  },
11  user: {
12    type: String,
13    required: true,
14  },
15  tangoDB: {
16    type: String,
17    default: '',
18  },
19  device: {
20    type: String,
21    required: true,
22  },
23  name: {
24    type: String,
25  },
26  value: {
27    type: Schema.Types.Mixed,
28  },
29  argin: {
30    type: String,
31  },
32  valueBefore: {
33    type: Schema.Types.Mixed,
34  },
```

(continues on next page)

(continued from previous page)

```
35     valueAfter: {  
36         type: Schema.Types.Mixed,  
37     },  
38 });
```