
ska-oso-ptt Documentation

Release 1.2.0

SKAO, (Trevor A Swain)

Feb 16, 2024

CONTENTS:

1	Search for Scheduling Block Instance (SBI)	3
2	Search for Scheduling Block Definition (SBD)	7
3	Search for Execution Block (EB)	11

The SKA OSO PTT is a tool currently under development which is made on top of react base skeleton present in SKA gitlab repository. Current PTT allows users to query SKA's OSO Data Archive for information on Scheduling Block Definition(SBD),Scheduling Block Instance(SBI) and Execution Block(EB) stored in the database and links between them (i.e. a certain SBI was generated from a certain SBD, and EB with EBID XX was derived from executing SBI YY) and display the results on the GUI.

This is a initial UI flow for Project tracking tool where user will be able to query ODA to view the details for SBI,SBD and EB. User will be able to fuzzy search by SBI ID, SBD ID and EB ID respectively. User will also be able to search SBI, SBD and EB by a Start Date and End Date. By default user will have quick filter for Last 7 days and Today records. It communicates with ska-db-oda via a REST API(see [the backend project documentation](#) for more details).

SEARCH FOR SCHEDULING BLOCK INSTANCE (SBI)

1.1 Search using Start Date and End Date

The screenshot displays the SKAO Project Tracking Tool interface. At the top, there are three search buttons: "SEARCH FOR SCHEDULING BLOCK INSTANCE (SBI)", "SEARCH FOR SCHEDULING BLOCK DEFINITION (SBD)", and "SEARCH FOR EXECUTION BLOCK (EB)". Below these, there are input fields for "Start Date" (17/02/2023) and "End Date" (21/01/2024), with a "SEARCH BY DATES" button. There is also a "Search by ID" field with a "SEARCH BY ID" button. Below the search filters, it says "Showing records for selected dates (Between 17-02-2023 and 21-01-2024)" with "Last 7 days" and "Today" filters. A table with 6 columns (Scheduling Block Instance ID, Scheduling Block Definition ID, Execution Block ID, Date Created, Created By, Telescope) contains 4 rows of data.

Scheduling Block Instance ID	Scheduling Block Definition ID	Execution Block ID	Date Created	Created By	Telescope
sbi-mvp01-20220923-00004	sbd-mvp01-20200325-00004	eb-mvp01-20200325-00001	25-11-2023	DefaultUser	ska_mid
sbi-mvp01-20220923-00005	sbd-mvp01-20200325-00003	eb-mvp01-20220923-00002	15-10-2023	DefaultUser	ska_mid
sbi-mvp01-20220923-00003	sbd-mvp01-20200325-00003	eb-mvp01-20220923-00002	14-10-2023	DefaultUser	ska_mid
sbi-mvp02-20220923-00001	sbd-mvp02-20200325-00002	eb-mvp02-20220923-00004	02-10-2023	DefaultUser	ska_mid

© SKAO 2024 | 1.1.0

1.2 Search using SBI ID with Fuzzy Search

SKAO | PROJECT TRACKING TOOL
⚙️

SEARCH FOR SCHEDULING BLOCK INSTANCE (SBI)

SEARCH FOR SCHEDULING BLOCK DEFINITION (SBD)

SEARCH FOR EXECUTION BLOCK (EB)

Start Date End Date

dd/mm/yyyy dd/mm/yyyy

Please enter start date Please enter end date

Search by ID

sbi-mvp01-20220923-000

Please enter ID

SEARCH BY DATES
SEARCH BY ID

Showing records for ID pattern "sbi-mvp01-20220923-00004"

Last 7 days
Today

Scheduling Block Instance ID	Scheduling Block Definition ID	Execution Block ID	Date Created	Created By	Telescope
sbi-mvp01-20220923-00004	sbd-mvp01-20200325-00004	eb-mvp01-20200325-00001	25-11-2023	DefaultUser	ska_mid

© SKAO 2024 | 1.1.0

1.3 Quick filter for Last 7 days and Today

SKAO | PROJECT TRACKING TOOL
⚙️

SEARCH FOR SCHEDULING BLOCK INSTANCE (SBI)

SEARCH FOR SCHEDULING BLOCK DEFINITION (SBD)

SEARCH FOR EXECUTION BLOCK (EB)

Start Date End Date

dd/mm/yyyy dd/mm/yyyy

Please enter start date Please enter end date

Search by ID

Please enter ID

SEARCH BY DATES
SEARCH BY ID

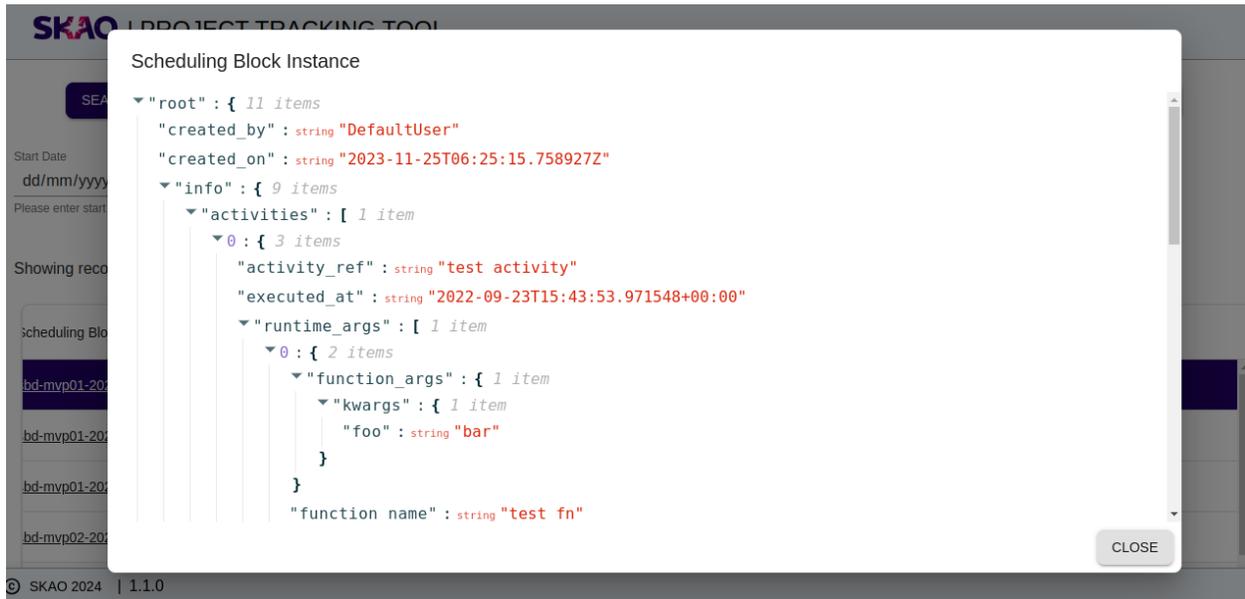
Showing records for last 7 days (between 15-01-2024 and 21-01-2024)

Last 7 days
Today

No SBI records found

© SKAO 2024 | 1.1.0

1.4 View Scheduling Block Instance



The screenshot displays the SKAO Project Tracking Tool interface. A modal window titled "Scheduling Block Instance" is open, showing a detailed view of a scheduling block instance. The data is presented in a structured, JSON-like format with expandable sections.

```
"root" : { 11 items
  "created_by" : string "DefaultUser"
  "created_on" : string "2023-11-25T06:25:15.758927Z"
  "info" : { 9 items
    "activities" : [ 1 item
      0 : { 3 items
        "activity_ref" : string "test activity"
        "executed_at" : string "2022-09-23T15:43:53.971548+00:00"
        "runtime_args" : [ 1 item
          0 : { 2 items
            "function_args" : { 1 item
              "kwargs" : { 1 item
                "foo" : string "bar"
              }
            }
          }
        "function name" : string "test fn"
```

The modal window includes a "CLOSE" button in the bottom right corner. The background shows a search bar and a list of scheduling blocks, with the selected block highlighted.

SEARCH FOR SCHEDULING BLOCK DEFINITION (SBD)

2.1 Search using Start Date and End Date

SKAO | PROJECT TRACKING TOOL 

[SEARCH FOR SCHEDULING BLOCK INSTANCE \(SBI\)](#) **[SEARCH FOR SCHEDULING BLOCK DEFINITION \(SBD\)](#)** [SEARCH FOR EXECUTION BLOCK \(EB\)](#)

Start Date:  End Date:  **SEARCH BY DATES**

Please enter start date Please enter end date

Search by ID: **SEARCH BY ID**

Please enter ID

Showing records for selected dates (Between 16-02-2023 and 22-01-2024) [Last 7 days](#) [Today](#)

Scheduling Block Definition ID	Execution Block ID	Date Created	Created By	Telescope	More Info
sbd-mvp01-20200325-00004	eb-mvp01-20200325-00001	14-08-2023	DefaultUser	ska_mid	
sbd-mvp01-20200325-00005	eb-mvp01-20200325-00001	17-08-2023	DefaultUser	ska_mid	
sbd-mvp01-20200325-00007	eb-mvp01-20200325-00001	16-08-2023	DefaultUser	ska_mid	
sbd-mvp01-20200325-00003	eb-mvp01-20220923-00002	15-08-2023	DefaultUser	ska_mid	

© SKAO 2024 | 1.1.0

2.2 Search using SBD ID with Fuzzy Search

SKAO | PROJECT TRACKING TOOL ⚙️

Start Date: |
 End Date: |

Search by ID: |

Please enter start date | Please enter end date | Please enter ID

Showing records for ID pattern "sbd-mvp01-20200325-00005"

Scheduling Block Definition ID	Execution Block ID	Date Created	Created By	Telescope	More Info
sbd-mvp01-20200325-00005	eb-mvp01-20200325-00001	17-08-2023	DefaultUser	ska_mid	

© SKAO 2024 | 1.1.0

2.3 Quick filter for Last 7 days and Today

SKAO | PROJECT TRACKING TOOL ⚙️

Start Date: |
 End Date: |

Search by ID: |

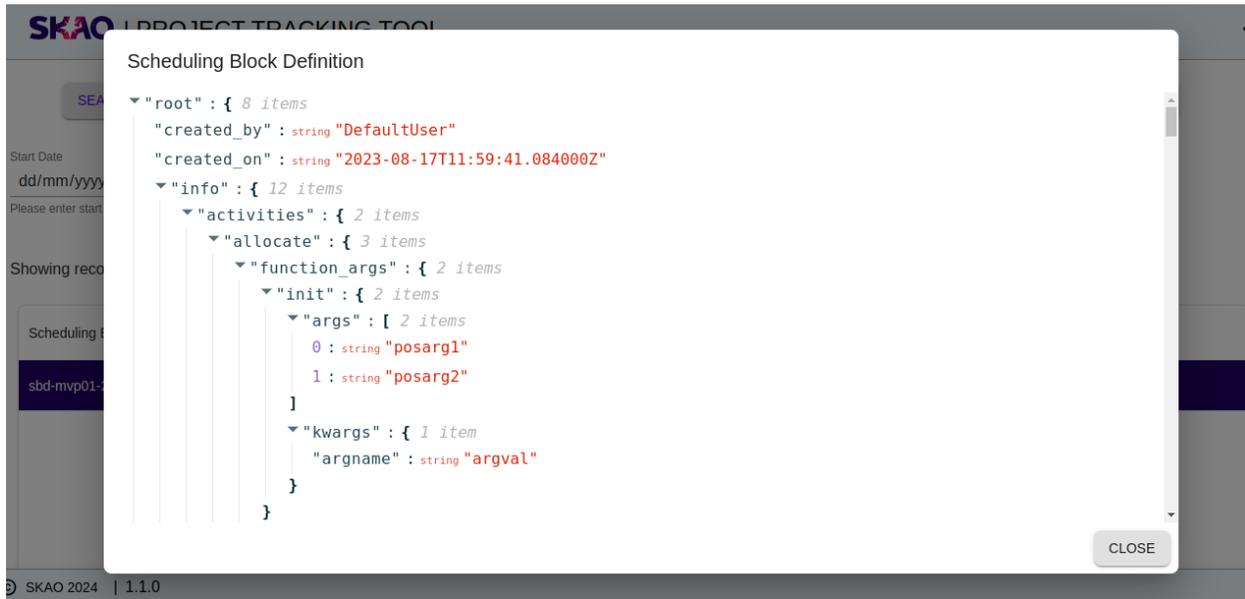
Please enter start date | Please enter end date | Please enter ID

Showing records for last 7 days (Between 15-01-2024 and 21-01-2024)

No SBD records found

© SKAO 2024 | 1.1.0

2.4 View Scheduling Block Definition



The screenshot displays the SKAO Project Tracking Tool interface. A modal window titled "Scheduling Block Definition" is open, showing a JSON-like structure of a scheduling block definition. The background interface includes a search bar, a date input field, and a list of scheduling blocks.

```
"root": { 8 items
  "created_by": string "DefaultUser"
  "created_on": string "2023-08-17T11:59:41.084000Z"
  "info": { 12 items
    "activities": { 2 items
      "allocate": { 3 items
        "function_args": { 2 items
          "init": { 2 items
            "args": [ 2 items
              0: string "posarg1"
              1: string "posarg2"
            ]
            "kwargs": { 1 item
              "argname": string "argval"
            }
          }
        }
      }
    }
  }
}
```

The modal window has a "CLOSE" button in the bottom right corner. The background interface shows the SKAO logo, a search bar, and a date input field with the text "Start Date dd/mm/yyyy". Below the date input, there is a section for "Showing records" with a table of scheduling blocks, including one with the ID "sbd-mvp01-".

SEARCH FOR EXECUTION BLOCK (EB)

3.1 Search using Start Date and End Date

SKAO | PROJECT TRACKING TOOL ⚙️

[SEARCH FOR SCHEDULING BLOCK INSTANCE \(SBI\)](#) [SEARCH FOR SCHEDULING BLOCK DEFINITION \(SBD\)](#) **[SEARCH FOR EXECUTION BLOCK \(EB\)](#)**

Start Date: End Date: **SEARCH BY DATES** Search by ID: **SEARCH BY ID**

Please enter start date Please enter end date Please enter ID

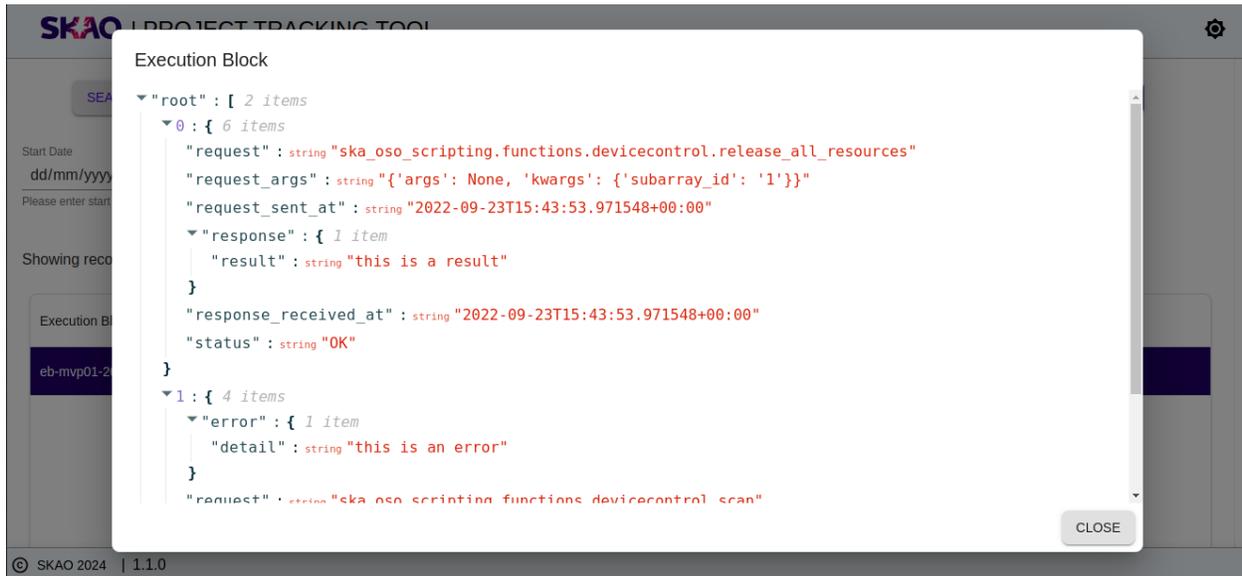
Showing records for selected dates (Between 13-04-2023 and 22-01-2024) [Last 7 days](#) [Today](#)

Execution Block ID	Scheduling Block Definition ID	Date Created	Created By	Telescope	More Info
eb-mvp01-20220923-00003	sbd-mvp01-20200325-00005	06-08-2023	DefaultUser	ska_mid	
eb-mvp01-20220923-00002	sbd-mvp01-20200325-00002	10-08-2023	DefaultUser	ska_mid	
eb-mvp01-20200325-00001	sbd-mvp01-20200325-00001	31-07-2023	DefaultUser	ska_mid	
eb-mvp01-20220923-00010	sbd-mvp01-20200325-00004	17-08-2023	DefaultUser	ska_mid	

© SKAO 2024 | 1.1.0

3.2 Search using EB ID with Fuzzy Search

3.4 View Execution Block



3.4.1 Requirements

This skeleton requires **Node** and **YARN** to install and run. To install these follow the instructions for your operating system at

[<https://nodejs.org/en/download/>](https://nodejs.org/en/download/).

Alternatively the official Node docker image can be used. Instructions can be found on the

[[official Node docker image site](https://github.com/nodejs/docker-node/blob/master/README.md#how-to-use-this-image)](https://github.com/nodejs/docker-node/blob/master/README.md#how-to-use-this-image).

3.4.2 Installation

Open the command prompt for your chosen environment._

1. Confirm Node and YARN are installed and configured correctly, both the following commands should return the relevant version number.

```
> node --version > yarn --version
```

2. Clone the project from GitHub
3. Allow yarn to be able to include required SKAO libraries

```
> npm config set @ska-telescope:registry https://artefact.skao.int/repository/npm-internal/
```

4. Install all the necessary project dependencies by running

```
> yarn init
```

5. Install required SKAO libraries

It is expected that required SKAO libraries would have been included at this point, however if this is found not to be the case, the following command will include them.

```
> yarn skao:update
```

```
### Steps to create a new release
```

Note, this does not currently work in the Windows Shell. Use either Linux, Mac, or Windows WSL.

Also note that the chart names should be updated when you use a different repo name.

Release

The following steps and commands is to create a new release for the portal.

1. Create a new branch from main branch.
2. Run one of `make bump-major-release`, `make bump-minor-release`, or `make bump-patch-release`
3. Update the `charts/ska-oso-ptt/values.yaml` file, the `image.version` should be updated.
4. Make sure the following files have the new version: `* charts/ska-oso-ptt/Chart.yaml` `* package.json`
`* .release`
5. Run `make git-create-tag`
6. Run `make git-push-tag`
7. You will then be able to merge that branch back in, and the new release should be created.

3.4.3 During development

It is note that the introduction of new libraries may throw an error. This is usually because WebPack requires the library to be included as part of the `ModuleFederationPlugin` entry within the `webpack.config.js` It is suggested that the new library be added into the area on the configuration annotated mixture.

Adjustment once final positioning within the SKA-Portal has been determined

So that there is no clashes with other applications originating from a skeleton, the following steps should be taken. Once these are done the application will no longer be available via the `ReactSkeleton` menu item within the SKA-Portal

1. `webpack.config.js` : Change the `devServer` port number from 8090.
2. `webpack.config.js` : Change the final `ReactSkeleton` entry to `NewApp`.
3. Relate these new values back to the developer responsible for updating the SKA-Portal

Once these steps have been completed, the application should be accessible from it's new location in the SKA-Portal

SKA Components

These are supplied from the various components. An overview of the libraries have been provided here for reference. For specifics of the components/functions available, please refer to the appropriate repository

ska-javascript-components : library containing a few pure components and types

ska-gui-components : library containing components that have been built using Material UI.

ska-gui-local-storage : library containing redux local storage. This is used by the ska-gui-components components as applicable

It is noted that all the components from ska-javascript are provided in addition as part of the ska-gui-components library, so separate inclusion is not required if the ska-gui-components are included.

Services

Included are a number of services, which have also been implemented into the code, providing simple examples. It is expected that in the main that there will be no updates to these services directly

Below is a list of the current services, together with their purpose.

i18n : Allows for text to be displayed in the language of the browser, with English as the default

theme : Contains the initialization of the latest SKAO Theme

3.4.4 Running and Building the Application

Scripts for running, testing, and building the application are provided as part of the standard configuration. These are run using YARN and listed in the scripts section of the package.json file.

From the project directory, you can run any of the following:

- `> yarn dev`
Runs the app in the development mode at [<http://localhost:8090>] (<http://localhost:8090>). The app will recompile and restart if you make any edits to the source files. Any linting errors will also be shown in the console.
- `> yarn skao:update`
yarn will update the repository with the latest SKAO libraries
- `> yarn start`
Same as `yarn dev` but for some implementations it is prefixed with `NODE_ENV=testing`. This is used in the CI/CD Processes
- `> yarn cypress`
Launches Cypress which has been set up to provide component testing. For further information on the use of Cypress, see <https://docs.cypress.io/guides/component-testing/overview>
- `> yarn test`
Launches the test runner in the interactive watch mode. See the `[testing](#testing)` section for more information.
- `> yarn build`
Builds the app for production to the `build` folder. The build is minified and any JSX is transpiled to JavaScript. Your app is ready to be deployed!

- `> yarn audit`

Checks the dependencies to see if there are any vulnerabilities.

The project will then be accessible at the url <http://localhost:8090/>

3.4.5 Testing

We use Cypress as the test running framework. It will look for test files within a number of locations, however the standard that the SKAO will employ will be the use of `.cy.{tsx | jsx}` in the same folder as the component being tested.

To run the interactive test runner, execute

```
> yarn test
```

This will also watch the source files and re-run when any changes are detected

To run the tests with coverage, execute

```
> yarn test:coverage:summary
```

The coverage results are displayed in the console. They are also written to the `coverage` folder.

- `./build/coverage/index.html` - open in a web browser to view

All the tests should pass before merging the code

Code Analysis

[ESLint](<https://ESLint.org/>) and [Prettier](<https://prettier.io/>) are included as code analysis and formatting tools. These do not need installing as they're included in `node_modules` by running `yarn init`. These tools can be run in the command line or integrated into your IDE (recommended).

To run the analysis tools, execute

```
> yarn code-analysis
```

This will display any errors in the command line. If there are any errors, YARN will exit with a non-zero code, the `-s` argument suppresses this and cleans up the output.