

API

- -
 - API
 -
 -
- - Uuid
 - filters
 - name
 - description
 - currentSubTabId
 - runUrlURL
 - externalApiUrlAPIURL
- - loadReport(options)
 - getCurrentSubTab()
 - getSubTabByPosition(position)
 - getSubTab(subTabId)
 - getSubTabFilters(SubTab) ()
 - nextSubTab()
 - previousSubTab()
 - findSubTabOrder(subTabId)
 - moveToTab(subTabId)
 - moveToTabAtPosition(position)
 - hasFilters()
 - refresh()
 - resetDashboard()
 - getToReport(repotUUID)
 - getEmbedURL()
 - getSubTabs()
 - getAllReports()
 - forEach(fn)
- - tabChanged

- -
 - ■
 -
 - API

API

ID2

API

API

API

DOM API

getAllReports()ID

- UUID - UUID
- UUID - ID
- UUID - UUID
- Id - ID

Yellowfin Yellowfin APInextSubTab()moveToTabAtPosition(4)

UUId

String

UUID

filters

API

API

name

String

description

String

currentSubTabIdID

Number

ID

runUriURL

String

```
http://localhost:8080/RunDashboard.i4?dashUUID=1e68d9cc-fa5a-44e2-816d-782aa40ceeae&primaryOrg=1&clientOrg=1
```

externalApiUriAPIURL

```
<script type="text/javascript" src="http://localhost:8080/JsAPI/v3?dashUUID=1e68d9cc-fa5a-44e2-816d-782aa40ceeae"></script>
```

loadReport(options)

Promise

promise

UUID/UUIDpromise

Options - Object

- **ReportId** - (String) UUIDUUID
- **EntityUUID** - ID

getCurrentSubTub()

```
let subTab = dashboard.getCurrentSubTab();
console.log(subTab.name);
```

getSubTabByPosition(position)

0

null

position: Number

```
let subTab = getSubTabByPosition(4);
if(subTab) {
  console.log(subTab.name);
}
```

getSubTab(subTabId)

IDID

IDnull

subTabId: String

```
//Get sub tab by Id
let subTab = dashboard.getSubTab(123456);
//Get subTab by name
let subTab = dashboard.getSubTab('Campaign Summary');
```

getSubTabFilters(SubTab) ()

Object - {String, [FilterObject](#)}

UUID

/

getFiltersUUIDgetAllFilters () [uuid] filters.getFilter (filterId) filters.forEach (fn)

nextSubTab()

```
document.querySelector('div#nextTabButton').addEventListener('click', function(e) {  
    dashboard.nextSubTab();  
});
```

10

```
setInterval(function() {  
    dashboard.nextSubTab();  
}, 10000);
```

previousSubTab()

```
document.querySelector('div#previousTabButton').addEventListener('click', function(e) {  
    dashboard.previousSubTab();  
});
```

10

```
setInterval(function() {  
    dashboard.previousSubTab();  
}, 10000);
```

findSubTabOrder(subTabId)

Number

null

subTabId: Number, String

ID

```
let subTabPosition = dashboard.findSubTabOrder('Campaign Summary')
```

ID

```
let subTabPosition = dashboard.findSubTabOrder(123456);
```

moveToTab(subTabId)

IDID

APItabChanged

subTabId: String

UUID

Campaign Summary

```
dashboard.moveToTab('Campaign Summary');
```

UUID28c595a5-c097-445c-ac9d-cbd4c415a667

```
dashboard.moveToTab('28c595a5-c097-445c-ac9d-cbd4c415a667');
```

moveToTabAtPosition(position)

APItabChanged

position: Number

0

```
dashboard.moveToTabAtPosition(0);
```

```
dashboard.moveToTabAtPosition(dashboard.getSubtabs().length - 1);
```

hasFilters()

Boolean

truefalse

```
let hasFilters = dashboard.hasFilters();
if(hasFilters) {
    console.log("this dashboard has modifiable filters");
}
```

refresh()

```
dashboard.refresh();
```

resetDashboard()

getToReport(repotUUID)

UUIDReport Can't be Found

reportUUID: String

UUID

getEmbedURL()

String

URL

externalApiUrl

getSubTabs()

Array[SubTab]

UUID

```
let subTabs = dashboard.getSubTabs();
subTabs.forEach(subTab => {
  console.log(subTab.name + " : " + subTab.uuid);
});
```

getAllReports()

Array[Object]

UUIDUUID

```
let reportToFind = 'af67e527-81d3-47fc-81ce-dfc506a61dd2';
let dashboardReports = dashboard.getAllReports();
let dashboardReport= dashboardReports.find(reportInfo => {
  return reportInfo.reportUUID === reportToFind;
});

if(dashboardReport) {
  //We found that report now we can load the report object. If the report has already been loaded by the dashboard this will just give us that version
  of the report
  dashboard.loadReport({
    reportId: reportInfo.reportUUID,
    entityUUID: reportInfo.entityUUID
  }).then(report => {
    //Now we've fetched the report.
  });
}
```

forEach(fn)

UUID

```
dashboard.forEach(subTab => {
  console.log(subTab.name + " : " + subTab.uuid);
});
```

tabChanged

- **currentSubTab**SubTab
- **currentSubTabId**NumberID
- **previousSubTabId**NumberID

```
dashboard.addEventListener('tabChanged', function(event) {  
  let newTab = event.currentSubTab;  
  console.log('We have just moved to the ' + newTab.name + ' tab');  
});
```

```
dashboard.addEventListener('tabChanged', function(event) {  
  let lastTab = dashboards.getSubTab(event.previousSubTabId);  
  console.log('Moved from ' + lastTab.name);  
})
```