

Basic Use

- [Overview](#)
 - [Enabling the Javascript API](#)
 - [Using the Javascript API](#)
 - [Basic Use](#)
- [Report URL Parameters](#)
 - [Examples](#)
- [Dashboard URL Parameters](#)
 - [Examples](#)

Overview

[top](#)

Enabling the Javascript API

The Javascript API is enabled on all new installations of Yellowfin, from version 5.1 onwards. If you have upgraded from an older version, contact [Yellowfin Support](#) for instructions on enabling the API for your installation.

Using the Javascript API

There are two ways to load a report or dashboard on your HTML page. The simple method involves including a single Javascript URL. The more advanced method allows you to customise the report or dashboard display and load content on demand from other scripts.

Basic Use

Include a script tag in your HTML page that points to your Yellowfin server. The report or dashboard identifier must be included as a parameter on the URL. For example:

```
<!-- report example -->
<script type="text/javascript" src="http://reporting.example.com/JsAPI?reportUUID=e5e5aaf3-c3b8-4f9b-8280-e21e4d848e63"></script>

<!-- dashboard example -->
<script type="text/javascript" src="http://reporting.example.com/JsAPI?dashUUID=3b0b6c9a-9dfb-41f0-b85a-eb17bb8aeeb9"></script>
```

The URL to the API is based on the URL used to access the Yellowfin front-end. For example, if users access Yellowfin through [*http://reporting.example.com:8080/*](#) then the Javascript API URL will be [*http://reporting.example.com/8080/JsAPI*](#). The URL will be accessed by the end user's browser, so it must be accessible from anywhere that end users will use your page (internal network, external, via VPN, etc.).

To ensure the server supports the version of the API your page is written for, you can use the `version` parameter. If the server supports multiple versions of the API, this parameter determines which version the server will use. If the server does not support the requested version, an error message will be displayed.

```
<script src="http://localhost/JsAPI?version=2.1&reportUUID=e5e5aaf3-c3b8-4f9b-8280-e21e4d848e63" type="text/javascript"></script>
```

If the `version` parameter is not included, the server will use its current version.

See [Version History](#) for more information.

Report URL Parameters

[top](#)

A report is identified by the `reportUUID` parameter. For backwards compatibility, it may alternatively be identified by the `reportId` or `wsName` parameters. Additional parameters may also be added to the URL, which will alter the way the report is displayed. The available parameters are:

Parameter	Description
reportUUID	Either <code>reportUUID</code> , <code>reportId</code> or <code>wsName</code> must be present. The unique ID identifying the dashboard to load.

reportId	Either reportUUID, reportId or wsName must be present. The numeric reportId identifying the report to load. It is recommended to use the reportUUID parameter instead.
wsName	Either reportUUID, reportId or wsName must be present. The Web Service name identifying the report to load. It is recommended to use the reportUUID parameter instead.
version	The API version to use. If the server does not support the requested version, an error message will be displayed. If this is not included, the server will use its current API version.
elementId	The id of the html element in which to load the report. If this is not included, a container element will be created at the point that the script tag is included.
showTitle	Default: true Set to false to omit the title bar at the top of the report. All interactive buttons included in the title bar will also be omitted.
showInfo	Default: true Set to false to omit the Info button in the title bar.
showFilters	Default: true Set to false to omit the Filters button in the title bar. Any user-prompt filters will not be displayed.
showSections	Default: true Set to false to omit the Sections button in the title bar (for reports with tabbed or multi-page sections).
showSeries	Default: true Set to false to omit the Series button in the title bar (for reports with the series selection option).
showPageLinks	Default: true Set to false to omit the previous page/next page button in the title bar (for reports with multiple pages).
showExport	Default: true Set to false to omit the Export button in the title bar.
height	Set this to a numeric value to override the report height.
width	Set this to a numeric value to override the report width.
display	Default: chart Set to table to display the report initially as a table. Set to chart to display the report initially as a chart. This is ignored for reports that do not have both table and chart available.
fitTableWidth	Default: true Set to true to attempt to scale the report to the width of the enclosing element.
canChangeDisplay	Default: true Set to false to omit the buttons that allow the user to switch between chart and table display.
username	Set this along with the password parameter to authenticate as a particular user when loading the report. This avoids the need for users to enter their login details before viewing restricted reports.
password	Set this along with the username parameter to authenticate as a particular user when loading the report.
token	Use a web service token to authenticate a user when loading the report.

Examples

This example includes a report and sets the dimensions of the report:

```
<script src="http://localhost/JsAPI?reportUUID=e5e5aaf3-c3b8-4f9b-8280-e21e4d848e63&width=500&height=350" type="text/javascript"></script>
```

This example uses an existing element to display the report in, and passes the user credentials:

```
<div id="myReportDiv" style="background: blue; width: 500px; height: 350px">
<script src="http://localhost/JsAPI?reportUUID=e5e5aaf3-c3b8-4f9b-8280-e21e4d848e63&elementId=myReportDiv&username=admin@yellowfin.com.au&password=test" type="text/javascript"><
/script>
</div>
```

This example uses an existing element to display the report in, overrides the report dimensions, and changes some of the initial display options:

```
<div id="myReportDiv" style="background: blue; width: 500px; height: 350px">
<script src="http://localhost/JsAPI?reportUUID=e5e5aaf3-c3b8-4f9b-8280-
e21e4d848e63&elementId=myReportDiv&width=400&height=300&showFilters=false&display=table" type="text/javascript"
></script>
</div>
```

Dashboard URL Parameters

[top](#)

A dashboard is identified by the `dashUUID` parameter. Additional parameters may also be added to the URL, which will alter the way the dashboard is displayed. The available parameters are:

Parameter	Description
<code>dashUUID</code>	Must be present. The unique ID identifying the dashboard to load.
<code>version</code>	The API version to use. If the server does not support the requested version, an error message will be displayed. If this is not included, the server will use its current API version.
<code>elementId</code>	The id of the html element in which to load the dashboard. If this is not included, a container element will be created at the point that the script tag is included.
<code>showTitle</code>	Default: true Set to false to omit the title bar at the top of the report. All interactive buttons included in the title bar will also be omitted.
<code>showInfo</code>	Default: true Set to false to omit the Info button in the title bar.
<code>showFilters</code>	Default: true Set to false to omit the Filters button in the title bar. Any analytical filters will not be displayed.
<code>showExport</code>	Default: true Set to false to omit the export button in the title bar.
<code>height</code>	Set this to a numeric value to override the dashboard height. If not specified, the dashboard height will be set large enough to show all reports without scrolling.
<code>width</code>	Set this to a numeric value to override the dashboard width. Set this to <code>auto</code> to expand to fill the parent element. If not specified, the dashboard width will be set based on the logged-in user's preference setting, or the system configuration setting.
<code>username</code>	Set this along with the <code>password</code> parameter to authenticate as a particular user when loading the dashboard. This avoids the need for users to enter their login details before viewing restricted dashboards.
<code>password</code>	Set this along with the <code>username</code> parameter to authenticate as a particular user when loading the dashboard.
<code>token</code>	Use a web service token to authenticate a user when loading the dashboard.

Examples

This example includes a dashboard, and sets the dimensions of the dashboard:

```
<script src="http://localhost/JsAPI?dashUUID=3b0b6c9a-9dfb-41f0-b85a-eb17bb8aeeb9&width=auto&height=500" type="text/javascript"></script>
```

This example uses an existing element to display the report in and passes user credentials:

```
<div id="myDashDiv" style="background: blue; width: 900px; height: 500px">
<script src="http://localhost/JsAPI?dashUUID=3b0b6c9a-9dfb-41f0-b85a-
eb17bb8aeeb9&elementId=myDashDiv&username=admin@yellowfin.com.au&password=test" type="text/javascript"></script>
</div>
```

This example uses an existing element to display the report in, overrides the report dimensions, and changes some of the initial display options:

```
<div id="myDashDiv" style="background: blue; width: 900px; height: 500px">
<script src="http://localhost/JsAPI?dashUUID=3b0b6c9a-9dfb-41f0-b85a-
eb17bb8aeeb9&elementId=myDashDiv&width=800&height=400&showFilters=false" type="text/javascript"></script>
</div>
```

[top](#)