



i-net
Clear Reports
2016

HTML Viewer Programming Guide

1 Introduction	2
1.1 Visual Customisation	2
1.2 Programmatic Customisation	2
1.2.1 HTML Viewer Running Sequence	2
1.2.2 Available Variables	3
1.2.3 Javascript Quickstart	5
1.2.4 Modifying the Menubar	5
1.2.5 Waiting for the Report to be finished	6

1 Introduction

The HTML Report Viewer can be customised on two different levels: visually and programmatically.

1.1 Visual Customisation

Though the viewer has its own internal visual styles it also derives colors from the theme provided for the overall system or the logged in user. The base colors therein will be used to have the HTML Report Viewer look like the rest of the interface which can use Bootstrap themes.

To customise even more, you can insert Less or CSS rules into any of your themes. The CSS selectors that you can use are straight forward and can be derived from the current viewer.

1.2 Programmatic Customisation

The HTML Viewer is extendable using the plugin architecture. You have to create a plugin and put your JavaScripts into it. Please see the samples for the `html_viewer` example. It provides the needed information to get a new custom plugin started.

Using the example you can add new JavaScripts at the end of the HTML Report Viewer. They will also be included in the exported Zip-File using the HTML format.

1.2.1 HTML Viewer Running Sequence

- **addInitEvent(<func>)**: adds function to a queue that will be processed right after all other scripts have been loaded. It is comparable to `jQuery(<func>)`.
 - The HTML Viewer will register using the function to start further execution.
 - It will then start to set up internal variable and the base URL to the called report. It will also initialise the keyboard handler
- **amIOnline.check(<func>)**: is another queue. It will check if the current viewer

is opened locally or using the report server. Only after using the queue it is safe to assume that the viewer is properly initialised.

- `check` takes a `function(<bool>){}` function, `bool` is a parameter indicating that the viewer uses the report server (`true`) oder is running locally.
- The Menubar will be set up now and the default zoom will be set
- The Tabbar will be set up and the report will be loaded asynchronously

1.2.2 Available Variables

Note: The following variables will be set up in the `init` phase of the viewer, triggered by `addInitEvent`. If you want to change them you will need to register your own `addInitEvent` function or put the `VARIABLES` based settings into the `PROMPT` global. It will be extended into `VARIABLES`.

Variable Name	Description	Default Value
BASE	The base URL to the report. This will be used for subsequent requests to the report server. Additional parameters should not be set here	
CANSHOWPERMALINK	Whether the export menu is allowed to show permalink for an export configuration.	<code>VARIABLES.canshowpermalink</code>
DRILLDOWNANDSUBREPORTSDISABLED		

Variable Name	Description	Default Value
DEFAULTZOOM	The Default Zoom that should be used. You can use XX% or Fit to Page. It will remember the last value used.	VARIABLES.defaultzoom or Fit To Page
HASNOEXPORTBUTTON	If the Viewer should have an export button	!VARIABLES.hasexportbutton
HASNOGROUPTREE	If the Viewer should have a group tree	!VARIABLES.hasexportbutton
HASNOPRINTBUTTON	If the Viewer should have a print button	!VARIABLES.hasprintbutton
HASNOPROMPTONREFRESH	If the Viewer should have button to show the prompts on a reload	!VARIABLES.hasexportbutton
HASNOTEXTSEARCH	If the Viewer should have the search button	!VARIABLES.hastextsearchcontrols
HASNOZOOM	If the viewer should have the zoom	!VARIABLES.haszoomcontrol
HASPROMPTS	If the viewer should display the prompts dialog	!VARIABLES.hasexportbutton
listener	This is the keyboard listener, globally registered	
PROMPT	Map of prompt values. This can be set from the report server as well	{}

Variable Name	Description	Default Value
PROMPTONREFRESH	If the Viewer should prompt on a refresh be default	!VARIABLES.promptonrefresh
TIMEOUTBEFOREERROR	Timeout before an error will be reported during loading a report file	10000
URI	The atomised parts of the URL that was called by the user in the browser	
VARIABLES	Map of variables set via the report URL.	{ }

1.2.3 Javascript Quickstart

The simplest script extension would be to do whatever you need directly in the script. But you have to expect that the HTML Viewer is not yet fully initialised and operational.

```
// This is probably the first thing you will see in the console
console.log('Hello World');

// This will be a blocking message pop-up before any of the viewer
initialisation has started
alert('Here I am.');
```

1.2.4 Modifying the Menubar

This will be the most common use case.

