

Basic Practices for Custom Scripts Developers

This article answers the following questions:

1. [Which built-in methods/properties do custom scripts have?](#)
2. [How to debug a custom script?](#)
3. [Where to find the information about the MI API?](#)

Most Used Custom Script Commands

Any Custom Script is represented by the **customScript** object that encapsulates a set of methods and properties. These methods and properties can be utilized to simplify the creation and usage of Custom Scripts. This section considers the most used of them in a more detail.

Methods

customScript.runApiRequest()

The `runApiRequest()` method utilizes jQuery's `ajax()` method.

It accepts two parameters:

- `url`: the URL to which the request is sent
- `settings`: a set of key/value pairs that configure the Ajax request; if no settings are passed, the default settings are used

Example: Obtain a list of all Datasets and their metadata by executing

```
customScript.runApiRequest(customScript.homeSite+'api/dataset');
```

customScript.close()

Finishes the script's execution.

Example: `setTimeout(function() { customScript.close(); },1000);`

customScript.log()

The screenshot shows the Metric Insights platform interface. At the top, there's a navigation bar with icons for Home, Custom Scripts / Slack Burst Script, Search, and user information (Admin, Robert). Below the navigation is a toolbar with tabs: Info (selected), Editor, Run History, Associated Schedules, and a set of actions (+, Q, 🗑, Saved). The main area is a code editor with line numbers from 22 to 44. Line 35 contains the code `customScript.log("Logging the custom script's message");`. A red arrow points from this line down to the 'Script Output' section below. The 'Script Output' section displays the message "Logging the custom script's message" in a yellow box. A 'Close Result' button is located in the bottom right corner of this section.

This method is an analogue of the `console.log()` method. The output is displayed in the **Script Output** section and on the *Run History* tab.

Example: `customScript.log("Logging the custom script's message");`

customScript.error()

The screenshot shows the Metric Insights interface for managing custom scripts. In the top navigation bar, 'Custom Scripts / Slack Burst Script' is selected. Below the navigation, there are tabs for 'Info', 'Editor' (which is active), 'Run History', and 'Associated Schedules'. On the right side of the editor area, there are buttons for '+ New...', 'Content', 'Admin', and a user profile for 'Robert'. The code editor displays a script with several lines of code, including line 36 which contains the command `customScript.error('Error message');`. A large black arrow points from this line down to the 'Script Output' section below. The 'Script Output' section shows the message 'Error message' in red text, indicating it was printed to the console. A yellow box highlights the word 'Error message' in the output. In the bottom right corner of the output section, there is a blue button labeled 'Close Result'.

Outputs an error that is displayed in the **Script Output** section and on the *Run History* tab.

Example: `customScript.error('Dataset with name
' + customScript.parameters.datasetName + ' is not found');`

Properties

customScript.parameters

`customScript.parameters.<parameter name>` is used to access [Custom Script's parameters](#), that were specified in the **Custom Script Editor**.

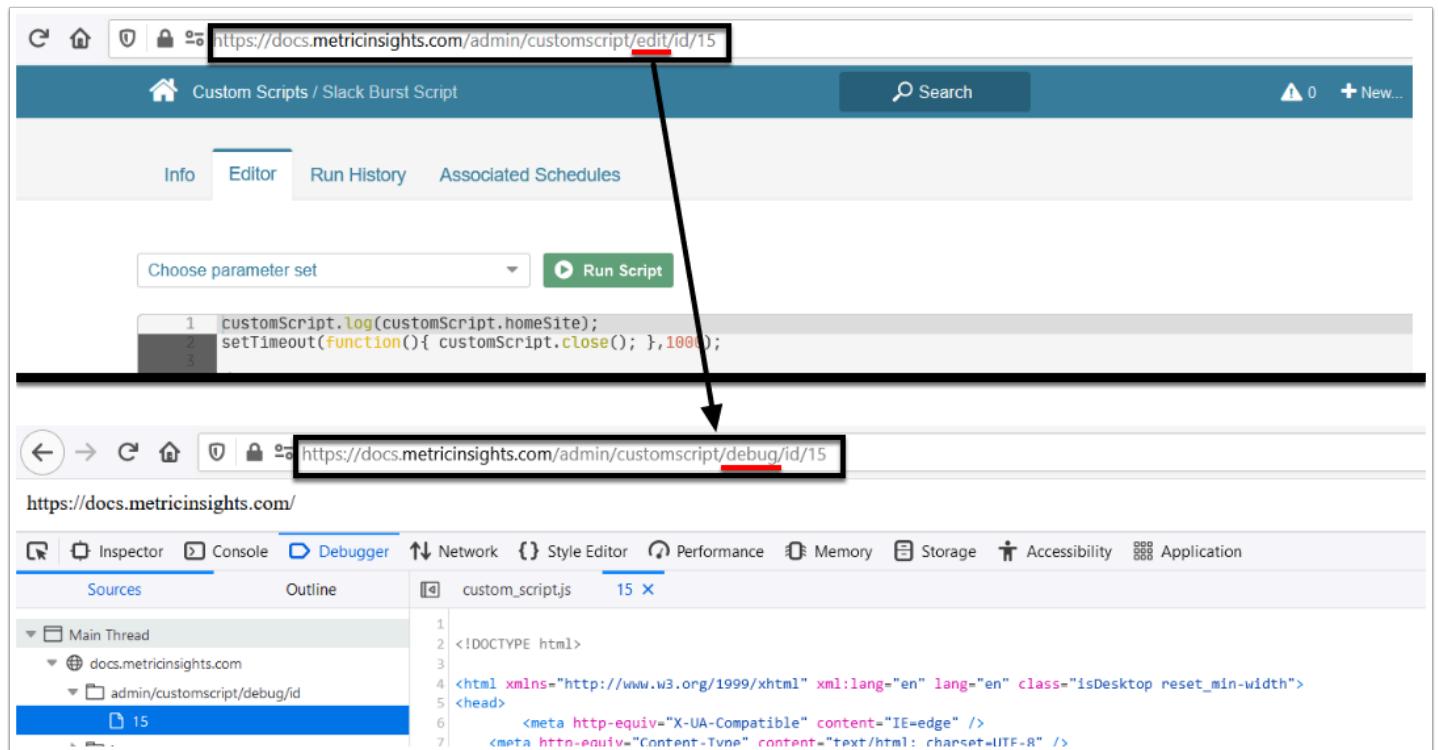
Example: `customScript.log(customScript.parameters.datasetName);` logs the script's `datasetName` parameter value

customScript.homeSite

The `customScript.homeSite` property holds the value of the MI instance's URL

Example: `customScript.log(customScript.homeSite);` logs the URL of the MI instance

Debugging Custom Scripts



To execute your Custom Script in the debug mode, simply replace `edit` in the script's URL with `debug` : `<hostname>/admin/customscript/edit/id/<script ID>` ?
`<hostname>/admin/customscript/debug/id/<script ID>`

Useful Links

- [Set up API Access](#): describes how to configure your Metric Insights instance for API Access.
- [Using API test tool](#): a chapter that explains how to test various API calls using API test tool
- [MI API calls](#): a chapter describing API calls for different MI elements
- [Examples of API calls in JavaScript](#): more examples of using the API in JavaScript