


String Loader

Block Group:	Data Services
Block Icon:	

Accesses the specified URL and either returns the string located in that file or performs other actions based on HTTP request methods.

The String Loader block is commonly used to load the contents of JSON or CSV files. You can link its output to a [JSON Parser](#) or [CSV Parser](#) block to create a table from the source file.

This block is also used to retrieve or post to HTML.

For information on using dataflow blocks, see [Dataflow](#).

Video Tutorial: Import CSV File

More video tutorials are [here](#).

Input/Output Properties

The following properties of the String Loader block can take input and give output.

- *invoke (trigger)*
- *enabled (boolean)*
- *interval (number)*
- *autoRun (boolean)*
- *timeout (number)*
- *method (enum or string)*
- *headers (string)*
- *sendCookies (boolean)*
- *path (project file path or URL)*
- *data (string or multi-line text)*

invoke causes the string to be loaded. The **invoke** property works only if **enabled** is set to TRUE.

enabled specifies whether the string loader is enabled.

- **TRUE**—The string loads as specified by **invoke**, **autoRun**, and **interval**.
- **FALSE**—The string does not load.

interval specifies how often the string is automatically reloaded, in seconds. A value of 0 means that the string is reloaded on **autoRun** or **invoke** only.

autoRun specifies whether the string is loaded automatically.

- **TRUE**—The string is loaded every time any property is changed. The first time that the **path** property is populated counts as a change.
- **FALSE**—The script runs only when the **invoke** property is triggered, or at the specified **interval**.

timeout specifies how long the block attempts to load the string before the request is canceled.

method specifies the [HTTP request method](#) for loading the string. In almost all cases, you want GET or POST.

- GET retrieves the string at the specified URL.
- POST adds something to the string at the specified URL.

headers sets the headers for the HTTP request. Use the following syntax:

```
<header1>=<value1>&<header2>=<value2>
```

NOTE: When using the POST method, specify the `Content-Type= application/x-www-form-urlencoded` header to ensure any variables you send are parsed correctly.

sendCookies specifies whether to send a cookie to the URL when retrieving the string.

There is a security risk involved when sending cookies. However, you must send cookies in order to get certain custom information, such as a particular user's Twitter feed.

path specifies the URL to load the string from. It can be a DGLux5 file path relative to the current project root, or it can be some other URL.

data sets and returns either the payload for a POST request, or the string to be added to the URL for a GET request.

Output Properties

The following properties of the String Loader block can give output but cannot take input.

- `loading` (*boolean*)
- `respHeaders` (*string*)
- `status` (*string*)
- `output` (*string*)
- `onComplete` (*event*)
- `error` (*string*)

loading returns whether the string is currently being loaded.

respHeaders returns the HTTP response headers received when the string is loaded.

status returns the [HTTP response status code](#).

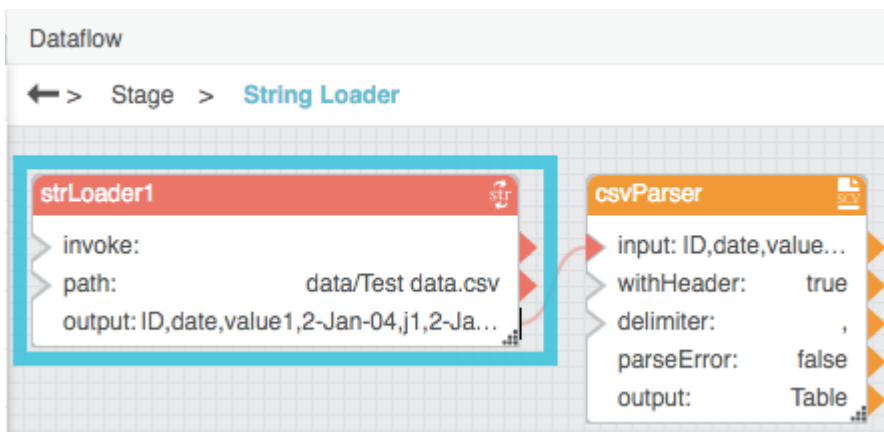
output returns the string contained in the file.

onComplete fires when the string finishes loading.

error returns the error message, if any.

Example

The following image demonstrates a typical use of the String Loader block. In this example, the String Loader block retrieves the contents of a CSV file. Then, the String Loader block passes the file contents to a CSV Parser block.



CORS Proxy

In the browser, the String Loader block cannot load data directly from another server unless that server defines a [CORS](#) header and allows other domains to access its data.

To load data from www.dglogik.com, simply use this path in your dataflow string loader block:
../p?http://www.dglogik.com

How to Add Rules to the Proxy for the DSA Server

DGLux Server has a built-in proxy that you can use to load data from a server that doesn't implement a

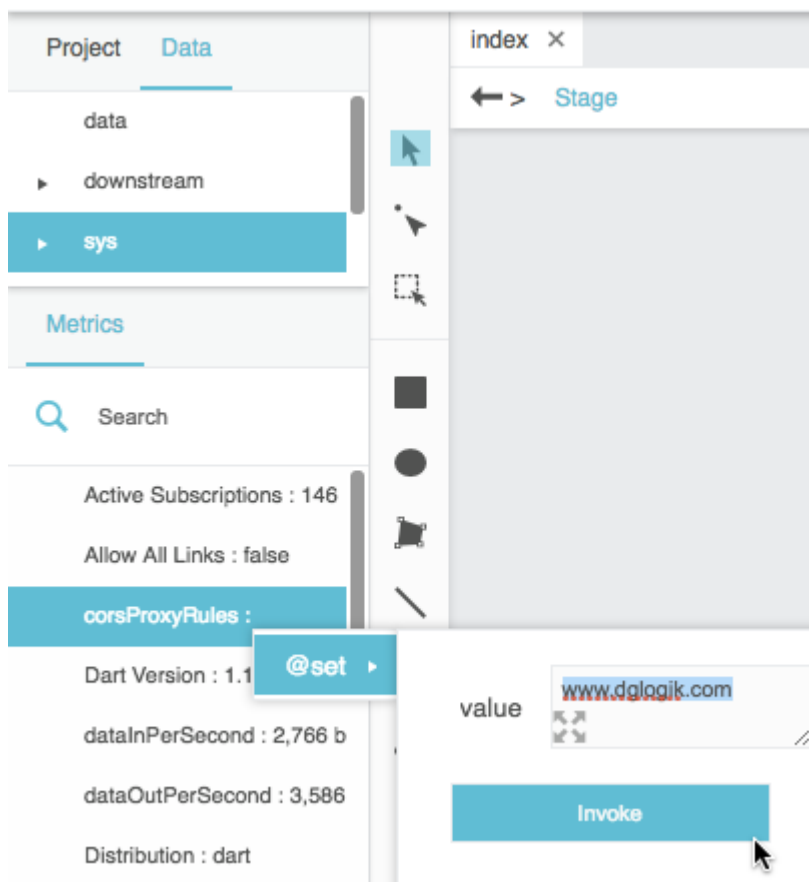
CORS header. (Not supported for server-side dataflows.)

Any proxy URL must be defined by a rule in the DSA point `/sys/corsProxyRules` before it can be used.

To configure a proxy rule:

1. In the [Data panel](#), choose `sys`.
2. In the [Metrics panel](#), right-click `corsProxyRules`.
3. Choose the `@set` action, and enter the list of rules. Rules are separated by new lines.

The following image demonstrates adding a proxy rule.



This is a whitelist rule list. By default, the list is blank, and nothing is allowed by the proxy.

The rules use [Regular Expressions](#) syntax.

Examples:

- `www.dglogik.com` : anything on the www.dglogik.com domain.
- `www.dglogik.com/company/about-dglogik$` : a strict match on the whole URL. Note this rule ends with a dollar sign (\$).
- `https://www.dglogik.com` : only allow https access, no http.
- `www.dglogik.com/company/about-dglogik(\?.+)?$` : strict URL, but allows different URL query strings to be used.

- .* : allow any URL. UNSAFE!

Additional notes about Regular Expression syntax:

- .* : match any character
- .+ : match at least one character
- \w+ : match any word that contains 0-9 a-z A-Z or _
- \$: match the end of the url, so additional character after the pattern it is not allowed
- if the rule doesn't start with http or https, a prefix `https?://` is automatically added, which matches both `http://` and `https://`

[Previous: List Node](#)

[Next: String Uploader](#)

From:
<https://wiki.dglogik.com/> - **DGLogik**

Permanent link:
https://wiki.dglogik.com/dglux5_wiki:dataflow:dataflow_blocks_reference:data_services:string_loader

Last update: **2021/09/20 15:03**

