



# DataScript: Functions

Avi Technical Reference (v18.1)

Copyright © 2019

# DataScript: Functions

[view online](#)

## Introduction

DataScripts are comprised of any number of function or method calls which can be used to inspect and act on traffic flowing through a virtual service. DataScript's functions are exposed via Lua libraries and grouped into modules: *string*, *vs*, *http*, *pool*, *ssl* and *crypto*. Other Lua libraries may also be used, following the documentation from [lua.org](http://lua.org). The following functions are available.

## DataScript

### String

[string.beginswith\( source, target \)](#)
[string.contains\( source, target \)](#)
[string.endswith\( source, target \)](#)
[string.len\( source \)](#)
[string.lower\( source \)](#)
[string.sub\( source, begin, \[end\] \)](#)
[string.upper\( source \)](#)

### VS

[avi.vs.client\\_ip\(\)](#)
[avi.vs.client\\_port\(\)](#)
[avi.vs.ip\(\)](#)
[avi.vs.log\(\)](#)
[avi.vs.name\(\)](#)
[avi.vs.port\(\)](#)
[avi.vs.rate\\_limit\( type, string\\_to\\_limit, \[defer\\_action=False\] \)](#)
[avi.vs.reqvar.\\*](#)
[avi.vs.table\\_insert\( \[table\\_name,\] key, value \[, lifetime\] \)](#)
[avi.vs.table\\_lookup\( \[table\\_name,\] key \[, lifetime\\_exten\] \)](#)
[avi.vs.table\\_refresh\( \[table\\_name,\] key \[, lifetime\\_exten\] \)](#)
[avi.vs.table\\_remove\( \[table\\_name,\] key \)](#)
[avi.utils.get\\_geo\\_from\\_ip](#)

### HTTP

[avi.http.add\\_cookie\( table \)](#)
[avi.http.add\\_header\( name, value \)](#)
[avi.http.close\\_conn\( \[reset\] \)](#)
[avi.http.cookie\\_exists\( name, \[context\] \)](#)
[avi.http.disable\(\)](#)

### Description

Search for string in beginning of a string

Search contains a string in another string

Search for string at the end of a string

Returns number of characters in string

Change a string to lower case

Extract a sub-string from a string

Change a string to upper case

Returns the client IP address

Returns the client source port

Returns IP address of the VS

Write a custom log to VS > client logs

Returns the name of the VS

Returns the VS port of the connection

Rate limits entities of various kinds.

Set a global variable usable across events

Store custom data in a time based table

Lookup data in a table

Update the expire time for a table entry

Remove data from a table

Used to extract geolocation database information

Insert a new cookie

Insert a new header and value

Close or reset a TCP connection

Validate if a cookie already exists

Upgrade (disable) HTTP processing for the current connection, which will subsequently be treated as layer 4 TCP

[avi.http.get\\_cookie\( name \[, context\] \)](#)[avi.http.get\\_cookie\\_names \( \[context\] \)](#)[avi.http.get\\_header\( \[\[name\] \[context\]\] \)](#)[avi.http.get\\_host\\_tokens\( \[start \[, end\]\] \)](#)[avi.http.get\\_path\( \[false\] \)](#)[avi.http.get\\_path\\_tokens\( \[start \[, end\]\] \)](#)[avi.http.get\\_query\( \[arg\\_name | avi.QUERY\\_TABLE\] \[, decode\] \)](#)[avi.http.get\\_req\\_body\(\[size\\_in\\_kb\]\)](#)[avi.http.get\\_reqvar\(\)](#)[avi.http.get\\_response\\_body\(size \[, offset\]\)](#)[avi.http.get\\_uri\( \[false\] \)](#)[avi.http.get\\_userid\(\)](#)[avi.http.hostname\(\)](#)[avi.http.internal\\_status\(\)](#)[avi.http.method\(\)](#)[avi.http.protocol\(\)](#)[avi.http.redirect\( uri \[,status\] \)](#)[avi.http.remove\\_cookie\( name1, \[name2, ...\] \)](#)[avi.http.remove\\_header\( name \)](#)[avi.http.replace\\_cookie\( table \)](#)[avi.http.replace\\_header\( name, value \)](#)[avi.http.response\( status, \[headers, \[body\]\] \)](#)[avi.http.scheme\(\)](#)[avi.http.secure\(\)](#)[avi.http.set\\_path\( new\\_uri \)](#)[avi.http.set\\_query\( integer | string | table \)](#)[avi.http.set\\_reqvar\(\)](#)[avi.http.set\\_response\\_body\\_buffer\\_size\(size\)](#)[avi.http.set\\_server\\_timeout\( time \)](#)[avi.http.set\\_uri\( new\\_uri \)](#)[avi.http.set\\_userid\(\)](#)[avi.http.status\(\)](#)

## Pool

[avi.pool.get\\_server\\_status\( pool, server, port \)](#)[avi.pool.get\\_servers\( pool \)](#)[avi.pool.select\( pool \[, server \[, port\]\] \)](#)[avi.pool.server\\_ip\(\)](#)

Return the values of a cookie

Return the names of cookies

Return header names or their values

Return a subsection of the host

Returns the URI's path /path.index.htm

Return a subsection of the path

Returns the URI's query ?a=1&amp;b=2

Returns part or all of the client request body.

Gets (reads) data stored in a variable via the avi.http.set() function.

Gets (reads) a specified number of kilobytes from the buffered response body starting at the specified kilobyte offset.

Returns the URI (path plus query)

Returns the user ID for the session

Return the hostname requested by client

Returns HTTP status as a string

Return the client's request method

Returns the session protocol, http or https

Redirect a request

Remove an existing cookie

Remove all instances of a header

Replace an existing cookies values

Replace an existing header's value

Send a defined HTTP response page

Returns http:// or https://

Returns on for https, nil for http

Modify the path of a request

Modify the query of a request

Sets (write) arbitrary data from an HTTP request event into a variable. These variables have scope across the HTTP\_REQ and HTTP\_RESP events.

Sets the maximum response body to be buffered.

Set a custom timeout value for a particular request.

Change the URI

Sets the user ID for the session

Returns status code to be sent to client

Returns the up/down status of the server

Returns up and total server count

Pick a specific pool

Returns the IP of the selected server

[avi.pool.get\\_server\\_info\(\)](#)

[avi.poolgroup.select\( pool \)](#)

## Groups

[avi.ipgroup.contains\( ipgroup, ip-address \)](#)

[avi.stringgroup.beginswith\( stringgroup, string \)](#)

[avi.stringgroup.contains\( stringgroup, string \)](#)

[avi.stringgroup.equals\( stringgroup, string \)](#)

[avi.stringgroup.endswith\( stringgroup, string \)](#)

## SSL

[avi.ssl.cipher\( \[true\] \)](#)

[avi.ssl.client\\_cert\( \[ \[avi.CLIENT\\_CERT\] \[, avi.CLIENT\\_CERT\\_FINGERPRINT\] \[, avi.CLIENT\\_CERT\\_SUBJECT\] \[, avi.CLIENT\\_CERT\\_ISSUER\] \[, avi.CLIENT\\_CERT\\_SERIAL\] \] \)](#)

[avi.ssl.protocol\(\)](#)

[avi.ssl.server\\_name\(\)](#)

## Crypto

[avi.crypto.decrypt\( ciphertext, key \[, iv \[, algo\]\] \)](#)

[avi.crypto.encrypt\( plaintext, key \[, iv \[, algo\]\] \)](#)

## Utilities

[avi.utils.base64\\_decode\( string \)](#)

[avi.utils.base64\\_encode\( string \)](#)

[avi.utils.murmur\\_hash \( string \)](#)

[avi.utils.sha1\\_hash \( string \)](#)

[avi.utils.md5\\_hash \( string \)](#)

Returns the IP address and port of the server

Pick a specific pool group

Compare an IP address against IPs within an IP Group

Compare a string against a list of strings within a string group

Compare a string against a list of strings within a string group

Compare a string against a list of strings within a string group

Compare a string against a list of strings within a string group

Return the SSL ciphers and settings

Returns the client's certificate, or part of it

Return the SSL version

Return SNI name field

Decrypt content

Encrypt content

Decode content

Encode content

Hash content

**Note:** Starting with Avi Vantage 17.1.10 and 17.2.3, DataScripts can hash arbitrary data using one of three functions supporting their corresponding named hashing methods: MurmurHash, SHA-1, and MD5.

For more information, refer to [DataScript: avi.utils.murmur\\_hash \(\)](#), [avi.utils.sha1\\_hash \(\)](#), [avi.utils.md5\\_hash \(\)](#).