



# DataScript Samples

Avi Technical Reference (v17.1)

Copyright © 2018

## DataScript Samples

[view online](#)

### Log Custom Data

Create a local log with custom data. This log is visible in the application logs, and is eligible to be sent to remote logging servers if the user-defined filter is configured in the virtual service's analytics profile. This logged data will be appended to standard log data that is generated for the request.

```
avi.vs.log("This is a custom log. The clients IP address is " .. avi.vs.client_ip())
```

### Persist on JsessionId

The following example looks at a server-generated cookie, using the value to persist the client to the server for an hour. This use case can also be performed via the native app-cookie persistence method.

```
-- Add to the HTTP Request
if avi.http.cookie_exists("JSESSIONID") and avi.http.get_cookie("JSESSIONID") ~= "" then
    var = avi.http.get_cookie("JSESSIONID")
    if avi.vs.table_lookup(var) then
        avi.pool.select(avi.vs.table_lookup(var))
    end
end

-- Add to the HTTP Response
if avi.http.get_cookie("JSESSIONID") ~= "" then
    avi.vs.table_insert(avi.http.get_cookie("JSESSIONID"), avi.pool.server_ip(), 3600)
end
```

### Redirect HTTP to HTTPS

If the client is accessing a secure portion of a website and is not using SSL, then redirect them HTTPS using a 302 Redirect.

```
if string.beginswith(avi.http.get_path(), "/secure") and avi.vs.port() ~= "443" then
    avi.http.redirect("https://" .. avi.http.hostname() .. avi.http.get_uri(), 302)
end
```

### Blacklist Enforcement of IP Addresses

Block clients coming from a designated blacklist group based on their client IP address. To implement this rule, first create an IP group, matching the name used in the following rule.

```
var = avi.vs.client_ip()
if avi.ipgroup.contains("IP-Group", var) then
```

```
avi.vs.log("Blacklisted IP" .. var.. "attempting to log in")
avi.http.close_conn()
end
```

## Content Switching ? Basic

This rule directs clients to a pool based on an exact match of the HTTP path.

```
if avi.http.get_path() == "/sales/" then
  avi.pool.select("sales-pool")
elseif avi.http.get_path() == "/dev/" then
  avi.pool.select("dev-pool")
elseif avi.http.get_path() == "/marketing/" then
  avi.pool.select("marketing")
end
```

## Content Switching ? Advanced

Direct clients to different pools, or different servers within a pool, based on a match against their path.

```
if string.startsWith(avi.http.get_path(), "/sales/") then
  avi.pool.select("sales-pool")
elseif avi.http.get_path() == "/engineering/" then
  avi.pool.select("engineering-pool", "apache1")
elseif avi.http.get_path() == "/marketing/" then
  avi.pool.select("marketing", "10.10.31.201")
end
```

## Content Switching ? Large List

For extremely long lists, it may be preferable to separate the DataScript code from the data it is using. This makes it easy to add or modify the list or mappings without giving users (or automation API calls) access to the underlying rule. This approach takes advantage of a string group, which contains the list and is referenced by the DataScript.

The string group must be set to key/value pair, with sample entries such as:

```
/marketing marketing-pool<br> /dev dev-pool<br> /checkout default-pool<br> /nonprod
nonprod-pool<br> /test nonprod-pool<br> /sales sales<br>
```

```
val, match = avi.stringgroup.startsWith("StringGroup", avi.http.get_path())
if match then
  avi.pool.select(val)
else
  avi.pool.select("default-pool")
end
```

## Rewrite Client Request

Add a query to the request based on the country code of the HTTP hostname. For instance, the site may be `www.avi.es/path/index.html?query=true`. The new request would read `www.avi.es/path/index.html?es=0123&query=true`

```

host = avi.http.host()
-- Create an array of mappings
TLD_map =

{es="123", fi="456", cn="789"}
if not host then
-- The host header does not exist, so close the connection
avi.http.close_conn()
elseif string.sub(host, 1, #"www.avi.") == "www.avi." then
i,j = string.find(host, "avi")
TLD = string.sub(host, j+2, -1)
new_query = "?mandate=" .. TLD_map[TLD]
old_query = avi.http.get_query()

if old_query > 0 then
new_query = new_query .. "&" .. old_query
end
avi.http.redirect("www.avi.com" .. avi.http.get_path() .. new_query, 302)
end

```

## Generate a different redirect URL, based on different host names

Check if the host name begins with `?www.avi.?`, then get the ccTLD from the host. Form the beginning part of the new query using the ccTLD map table to get the value of the query argument `?mandate?`. Append the old query, if it exists, to the new query. Generate the HTTP redirect to the new URL using the domain `?www.avi.com?` and the new query.

```

local ccTLD_map = {es="01345F", fi="5146FF", cn="45DFX", io="123456", ly="ABC123"}
host = avi.http.host()

if not host then
avi.http.close_conn()
elseif string.sub(host, 1, #"www.avi.") == "www.avi." then
i,j = string.find(host, "avi")
ccTLD = string.sub(host, j+2, -1)
new_query = "?mandate=" .. ccTLD_map[ccTLD]
old_query = avi.http.get_query()

if #old_query > 0 then
new_query = new_query .. "&" .. old_query
end
avi.http.redirect("www.avi.com" ..

```

```
avi.http.get_path() ..  
new_query, 302)  
end
```

## Rewrite publicly available hosts & URLs to private hosts and URLs &dash; which are in fact OpenShift routes

This can be a common requirement for OpenShift environments, since routes are not typically open and accessible directly from the Internet.

```
host = avi.http.get_header("Host")  
path = avi.http.get_path()  
if string.contains(host, "rhmap.global.acme.com") and string.beginswith (path, "/router/test") then  
    avi.vs.log("router/test match")  
    avi.http.replace_header ("Host", avi.http.get_path_tokens(3,3) ..".."apps.test-a.0341.o2.wel.cs1.cd2.acme.com")  
    avi.vs.log ("router/test new header")  
    if avi.http.get_path_tokens(4) then  
        avi.http.set_path ("/"..avi.http.get_path_tokens(4))  
    else  
        avi.http.set_path ("/")  
    end  
elseif string.contains(host, "rhmap.global.acme.com") and string.beginswith (path, "/router/dev") then  
    avi.vs.log("router/dev match")  
    avi.http.replace_header ("Host", avi.http.get_path_tokens(3,3) ..".."apps.test-a.0341.o2.wel.cs1.cd2.acme.com")  
    avi.vs.log ("router/dev new header")  
    if avi.http.get_path_tokens(4) then  
        avi.http.set_path ("/"..avi.http.get_path_tokens(4))  
    else  
        avi.http.set_path ("/")  
    end  
elseif string.contains(host, "rhmap.global.acme.com") and string.beginswith (path, "/core") then  
    avi.vs.log("core match")  
    avi.http.replace_header ("Host", "rhmap.apps.test-a.0341.o2.wel.cs1.cd2.acme.com")  
    avi.vs.log ("core new header")  
    avi.http.set_path ("/"..avi.http.get_path_tokens(2))  
else  
    avi.vs.log ("nothing matches")  
end
```