

20 Lines or Less #60: Body Modification, Balancing Acts and Awareness



Colin Walker, 2012-08-08

What could you do with your code in 20 Lines or Less? That's the question I like to ask for the [DevCentral](#) community, and every time I go looking to find cool new examples that show just how flexible and powerful iRules can be without getting in over your head.

This morning I had the distinct pleasure of presenting a new iRules Challenge to an upcoming group of FSEs here in Seattle for some intense training. I'll write more about that later, but it brought up an interesting question. After I had presented the requirements for the desires iRules solution to the group, one of the questions was, "About how long are iRules, generally speaking? a couple hundred lines? Closer to a thousand?". I can understand the question and why someone might jump to those sort of numbers. iRules can do some really cool, powerful stuff. Certainly in my previous life as a Perl guy it was entirely common to live in the triple digits for even relatively simple scripts, and break into multi thousand line programs when building daemons or things that were overly complex. That being said, it made me grin, because iRules are so wonderfully simple that we often only need a few lines to work some application magic.

I told the group about this series, that I've been doing for years now, off and on, and they were impressed that there were so many examples of useful, interesting iRules that take less than 21 lines to express. I have to say, in that moment I stopped and reflected and found the same to be true. It's darn impressive what you can do with an iRule in only 20 lines of code, and I had no idea that when I started this series I'd be continuing to explore those possibilities this far down the path. After many versions, syntax changes, command iterations and even DC upgrades, there are still so many more things that can be done to show off iRules in 20 Lines or Less. So, this week I've got some more for you, and I hope you enjoy.

HTTP Body Rewrite

<http://bit.ly/O5zY8k>

One of the topics that has come up time and time again, whether on the road giving presentations or in the forums, is content re-writing. It's not that it's difficult, per se, it's just that it's a common topic and not everyone knows exactly how to go about setting this up via an iRule. It used to be more complex and alchemical, but these days it's far more straight forward. Michael Yates, DC MVP, walks through precisely how to make use of this handy functionality in a response to this post. If you've ever needed to modify a URL, protocol prefix, domain, or anything else in an HTTP response, this is the post to bookmark.

```
1: # Example which replaces http:// with https:// in response content
2: # Prevents server compression in responses
3: when HTTP_REQUEST {
4:
5:     # Disable the stream filter for all requests
6:     STREAM::disable
7:
8:     # LTM does not uncompress response content, so if the server has compression enabled
9:     # and it cannot be disabled on the server, we can prevent the server from
10:    # sending a compressed response by removing the compression offerings from the client
11:    HTTP::header remove "Accept-Encoding"
12: }
13: when HTTP_RESPONSE {
14:
15:     # Check if response type is text
16:     if {[HTTP::header value Content-Type] contains "text"}{
17:
18:         # Replace http:// with https://
19:         STREAM::expression {@http://@https://@}
20:
21:         # Enable the stream filter for this response only
22:         STREAM::enable
23:     }
24: }
```

<http://bit.ly/Rra54L>

We've seen a couple different solutions to the "what do I do with my mobile requests?" question grace the 20LoL. This is another permutation and one that I think has merit. Being able to quickly and easily redirect users to a mobile version of the site via a query string seems like a solid methodology for many applications. I must say that I'd prefer to see the use of the class command rather than matchclass, but for people on versions older than 10.1 I suppose that isn't an option. For those of you on more recent versions, definitely make that change before deploying as you'll see considerable performance improvements. Other than that though, that's a very handy concept for those trying to determine how to differentiate mobile users to their application.

```
1: when HTTP_REQUEST {
2:   if { !([HTTP::uri] contains "requestAgent") } {
3:     if { ([matchclass [HTTP::header "User-Agent"] contains $::MobileAgents]) } {
4:       HTTP::redirect "https://[HTTP::host][HTTP::uri]?requestAgent=mobile"
5:     }
6:   }
7: }
```

Load Balancing to a Domain

<http://bit.ly/OQt0pm>

Load balancing and routing to pools or pool members is about as bread and butter as you can get when passing traffic through an ADC. Sure, we load balance, but that's not exactly the cool stuff anymore. How about a twist on load balancing, though? What if you want to be able to load balance to a given host name, rather than an IP address? What if every time a new connection comes in, you want to perform a DNS query to look up the current IP of a given domain, then send the traffic there? Now that's a little bit more interesting, I'd say. Also, with iRules? It's super easy. Michael Yates, DC MVP extraordinaire is back on the case for this example, worth a look see.

```
1: # Select the first returned IP address as the destination IP (inherits the destination port from the client's destination port).
2: when RULE_INIT {
3:   set static::dns_vs my_dns_vs
4: }
5:
6: when CLIENT_ACCEPTED {
7:
8:   # Get IP(s) for hostname www.example.com against 4.2.2.1 name server
9:   set ips [RESOLV::lookup @$static::dns_vs -a "www.example.com"]
10:
11:  # Log result. If there are multiple IP's it could be a TCL list like {1.1.1.1 2.2.2.2 3.3.3.3}.
12:  log local0. "Looked up www.example.com and found $ips, parsed first element: [lindex $ips 0]"
13:
14:  # Check if the first list element was empty
15:  if {$ips eq ""}{
16:    # Input wasn't an IP address, take some default action?
17:  } else {
18:    # Select the IP
19:    node [lindex $ips 0]
20:  }
21: }
```

And that brings this edition of the 20 Lines or Less to a close. With nearly 200 entries, the next few installments are sure to be good ones. Check back in two weeks for more iRules goodness steeped in brevity.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113