

v11 iControl: Sessions



George Watkins, 2011-01-09

Introduction

Version 11 introduces the concept of iControl sessions. iControl sessions are a stateful set of attributes (at this time, active folder and transaction) that persist across multiple requests for a single user. This allows a user to set remote session attributes on the BIG-IP and reuse them in subsequent requests. In addition, it also segregates iControl requests from other clients using the same credentials.

An iControl session by default is identified only by the user making the request. By default these requests act on the '/Common' folder and have a timeout of 30 minutes. Therefore if you continue using iControl as you have been you are actually activating a session, but it is completely transparent. This works well until there are requests initiated concurrently by the same user that depend on attributes set on the BIG-IP. If the iControl script is using these attributes it can make for some chaotic communication with the BIG-IP. In order to provide more granularity we can use a unique session identifier in our requests. It is from within one of these unique sessions that we can initiate a transaction and begin sending iControl requests to the BIG-IP.

Session Identifier

A session identifier is just a unique integer (think primary key here) that when inserted into a requests prevents multiple concurrent requests from the same user from interfering with each other. The session identifier is retrieved via a method called 'get_session_identifier' in the [System::Session interface](#). It can then be inserted into subsequent requests via an HTTP or SOAP header. The HTTP header is called 'X-iControl-Session' and has a value equal to that returned from the 'get_session_identifier' method. The SOAP header is simply called 'session' and uses the same integer. Both are equally acceptable methods and if there is a issue with one, the other will automatically be used.

Most SOAP clients provide an easy mechanism for inserting one or both of these. It is possible for this value to be forged, but it would require an individual knowing the user's credentials and intentionally manipulating another individuals requests.

Using A Unique Session To Create A Pool

We will be using Ruby for our example today. For those of you who are unfamiliar, the iControl Ruby Library is one of a number of [iControl libraries](#) (clients) we offer. Check out our [Getting Started With Ruby and iControl](#) Tech Tip to get Ruby up and running. Even if Ruby's not your thing, we offer a [number of other clients](#) and the concepts and examples outlined here apply to those as well.

In this example we are going to create a new pool using an iControl session identifier. Here are the steps for creating a new pool with a unique session:

1. Initiate our SOAP connection
2. Request a session ID
3. Assign it to our interfaces via an HTTP header
4. Set our timeout to 5 minutes
5. Our active folder to '/Common'
6. Create the pool using our session.

And, here's the code...

```
#!/usr/bin/ruby
```

```

require 'rubygems'
require 'f5-icontrol'

# initiate iControl interfaces

bigip = F5::IControl.new('10.0.0.1', 'admin', 'admin', \
  ['System.Session', 'LocalLB.Pool']).get_interfaces

# request a session ID

session_id = bigip['System.Session'].get_session_identifier

# assign the session ID to our interfaces via an HTTP header

bigip.each_value do |interface|
  interface.options['protocol.http_custom_headers'] = { 'X-iControl-Session' => session_id }
end

# set the timeout to 300 seconds and the active folder to /Common

```

```

bigip['System.Session'].set_session_timeout(300)
bigip['System.Session'].set_active_folder('/Common')

# any actions performed beyond this line of the code will be subject to session attributes

puts "Creating my_new_http_pool using session_id #{session_id}..."

bigip['LocalLB.Pool'].create_v2( \
  ['my_new_http_pool'], \
  ['LB_METHOD_ROUND_ROBIN'], \
  [ [ {'address' => '10.0.0.100', 'port' => 80 } ] ] )

```

While it may not seem like this code does a whole lot beyond just creating a pool, it ensures that multiple clients making concurrent changes to the BIG-IP do not step on each others feet. This fixes a large issue whereby many systems are performing changes to the BIG-IP in tandem and constantly stepping on each others feet. This also allows for the concept of changing folders within the BIG-IP, which is also a concept central to version 11. If there is a very large folder structure on a BIG-IP it can become tedious referring to every object by its absolute path.

Conclusion

iControl sessions are huge feature within version 11 and pave the way for the next Tech Tip in the v11 iControl series: Transactions. Without the concept of a session, it would be difficult to discern which requests were destined for a particular transaction versus another or whether they should be applied in real-time. In the Transactions Tech Tip we will dig deep into their behavior and possible implementations for a transaction. See everyone next week!

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