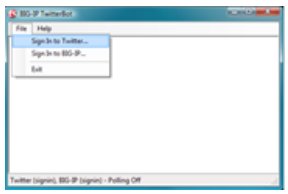
Two TwitterBots Are Better Than One!

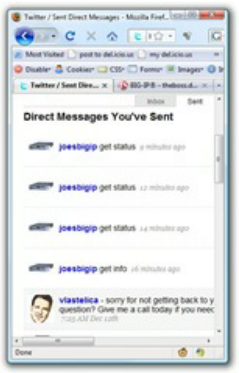
Joe Pruitt, 2008-15-12



I've been working on this for a while but never got around to finishing it. Lori's post from today reminded me of my "unfinished" work so I dug it up and brought it back to life.

A while back, I wrote a Twitter Proxy for BIG-IP's that used the iControl event API to listen of configuration change events to occur and then post those events to a pre-configured Twitter account.

A longer time ago, I wrote a IM bot that allowed you to "chat" with your BIG-IP but that relied on a 3rd party service that is no longer around so the app isn't much use to anyone. But, with that in mind, I thought to myself that Twitter since have a published API, it should be relatively easy to build a client that checked on replies and direct messages and then send replies back to the requesting user.



So I set off to see if there was a .Net client library for the Twitter API and was surprised to find out that there wasn't a complete one (at least that I could find). The ones that I found had a few calls supported (mainly in the Status group of messages) but I wanted to have something more complete. Twitter has methods for Status, User, Direct Messages, Friendship, Account, Favorite, Notification, Block, and Help so I set out to build the library myself. I ran into a bunch of hurdles due to the inconsistencies in their REST implementation and the dynamically changing behavior of the methods over time (see the Changelog for examples). I'll highlight these in a future article about the TwitterAPI .Net assembly project.

That brings me to my .Net TwitterBot. Using the TwitterAPI library I wrote, I created a client application that allows you to specify a Twitter account and a BIG-IP account you would like to interact with. The program will then monitor both reply and direct messages sent to the specified twitter account. It will then process the request, perform the relative iControl operations, and then generate a 140 character response to the user via a direct message. I opted to go this route for security reasons. Since direct messages are private for the receiving user, sensitive information will not accidentally be published to the general public.

As of this point, there are only two commands supported. "get info" will return system information for the device such as the serial number, product info, and system uptime. The "get status" method will make status calls for all the pool objects and return warnings for pools that are marked as down. With a little thought, it would be very easy to expand this command set to make this app a full command proxy for managing your device.

I've released the source for this app, including the .Net Twitter client API library as a alpha labs project. If it gains any traction, I'll make it more official and build a labs project page around it.

So, thanks Lori for your PHP TwitterBot. I always like a little internal competition and hopefully having two versions of this type of application will force us to both expand the features to keep up with each other and ultimately make both programs better. I'd love to hear feedback and if anyone is willing to expand on the functionality, or fix some bugs B-), please let me know and I'll patch them into the source distribution here.

Click Here to Download the BIG-IP TwitterBot Source

Happy Twittering!

-Joe

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