IRC HACKS™

100 Industrial-Strength Tips & Tools

O'REILLY®

Paul Mutton

HACK #51 Search the Web with Google

Performing Google searches from IRC is not only convenient, but also efficient. See how fast you can Google for something on IRC and click on the URL highlighted by your IRC client.

When someone pops into your IRC channel with a question, you can bet your life that 9 times out of 10, he could have easily found the answer on Google. If you think this is the case, you could tell him that—or you could do it slightly more subtly by suggesting a Google search term to an IRC bot, which will then go and look for a result.

Most IRC clients are capable of highlighting URLs in channels. Clicking on a highlighted URL will open your default web browser and load the page. For some people, this is a lot quicker than finding the icon to start your web browser and then typing or pasting the URL. More obviously, a single Google search will present its result to everybody in the channel.

Google Web APIs

Searching Google from within your own application is very easy, thanks to the Google Web APIs. The developer's kit (*http://www.google.com/apis*) contains a Java library that provides a wrapper around the Google Web API's SOAP interface. This essentially means that you can use it to perform Google searches from a Java IRC bot.

You will also need to create a Google account and obtain a license key. As I write this, the free license key entitles you to 1000 automated queries per day. This is more than enough for a single IRC channel.

You'll find oodles of Google goodness and more Google Web API hacking in *Google Hacks* by Tara Calishain and Rael Dornfest (O'Reilly).

The *googleapi.jar* file contains the classes that will be used by the bot when it performs Google searches, so you will need to make sure this is in your classpath when you compile and run the bot.

The goal is to have an IRC bot called GoogleBot that responds to the `!google` command. It will respond by showing the title and URL of the first Google search result. If the size of the page is known, that will also be displayed.

Your license key will be a simple String, so you can store that in the GoogleBot class as `googleKey`.

The Code

Create a file called *GoogleBot.java*:

```
import org.jibble.pircbot.*;
import com.google.soap.search.*;

public class GoogleBot extends PircBot {

    // Change this so it uses your license key!
    private static final String googleKey =
"OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO";

    public GoogleBot(String name) {
        setName(name);
    }

    public void onMessage(String channel, String sender, String login,
            String hostname, String message) {

        message = message.toLowerCase().trim();
        if (message.startsWith("!google ")) {
            String searchTerms = message.substring(8);

            String result = null;
            try {
                GoogleSearch search = new GoogleSearch();
                search.setKey(googleKey);
                search.setQueryString(searchTerms);
                search.setMaxResults(1);
                GoogleSearchResult searchResult = search.doSearch();
                GoogleSearchResultElement[] elements =
                        searchResult.getResultElements();
                if (elements.length == 1) {
                    GoogleSearchResultElement element = elements[0];
                    // Remove all HTML tags from the title.
                    String title = element.getTitle().replaceAll("<.*?>",
"");
                    result = element.getURL() + " (" + title + ")";
                    if (!element.getCachedSize().equals("0")) {
                        result = result + " - " + element.getCachedSize();
                    }
                }
            }
            catch (GoogleSearchFault e) {
                // Something went wrong. Say why.
                result = "Unable to perform your search: " + e;
            }

            if (result == null) {
                // No results were found for the search terms.
```

```
                result = "I could not find anything on Google.";
            }

            // Send the result to the channel.
            sendMessage(channel, sender + ": " + result);
        }
    }

}
```

You now need a main method to tell the bot which channels to join. If you want, you can tell the bot to join more than one channel, but remember you are limited in the number of Google searches you can do per day.

Create the file *GoogleBotMain.java*:

```
public class GoogleBotMain {

    public static void main(String[] args) throws Exception {
        GoogleBot bot = new GoogleBot("GoogleBot");
        bot.setVerbose(true);
        bot.connect("irc.freenode.net");
        bot.joinChannel("#irchacks");
    }

}
```

Running the Hack

When you compile the bot, remember to include both *pircbot.jar* and *googleapi.jar* in the classpath:

```
C:\java\GoogleBot> javac -classpath .;pircbot.jar;googleapi.jar *.java
```

You can then run the bot like so:

```
C:\java\GoogleBot> java -classpath .;pircbot.jar;googleapi.jar GoogleBotMain
```

The bot will then start up and connect to the IRC server.

The Results

Figure 8-1 shows GoogleBot running in an IRC channel and responding with the URL, title, and size of each of the results of a Google search.

Performing a Google search is a very popular task for bots to do. Take this into account if you run your bot in a busy channel, as there may be a bot there that already lets users search Google.

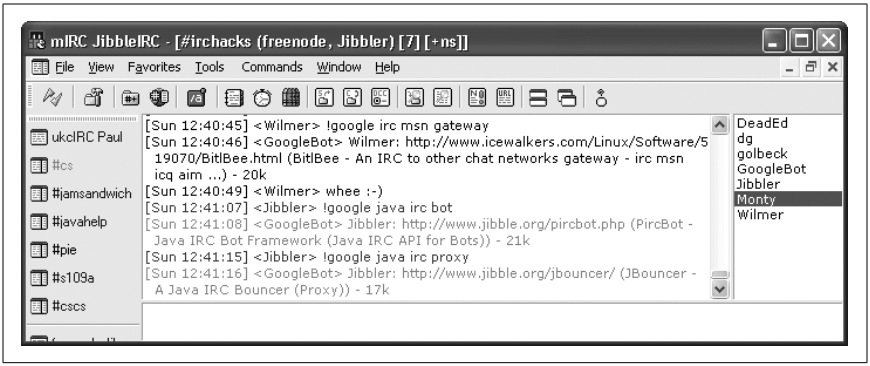

Figure 8-1. GoogleBot performing an IRC-related search