Impact of Internet on Business Activities in Serbia and Worldwide

Uticaj Interneta na poslovanje u Srbiji i svetu

DOI: 10.15308/SINTEZA-2014-863-867

INTERNET ROBOTIC SOFTWARE – POTENTIAL AND APPLICATION

Vladimir Stanojević, Mladen Veinović
Singidunum University, Serbia

Abstract:
Internet robotic appliactions, also known as „bots", have a broad sprectrum of usage. Bots are growing in numbers exponentially, following the expansion of Internet and it's massive usage. Best known bots are for example Google search bots, which access internet pages content, browse the data, and sends selected data in appropriate form to central database, where that data get categorised and organised for indexing purposes. This enables effective search and reporting results of a search by some terms. However, bots can have much more sophisticated applications: To replace living person in some simple or complex operations, both for purpose of acquiring wanted data, or update/insert some data in behalf of some business system/software. This way that operations goes faster, more efficient, and without human error factor. This makes bots very actuel segment of internet software, both for users and programers. In this paper, potential of bots is illustrated on example of automatic access, search and presenting multilayered, heterogenous data of business subjects in Republic of Serbia, regarding public available data from databases of APR and NBS.

Key words:
internet robots,
bots,
database,
APR,
NBS.

INTRODUCTION

Internet robotic applications, are specific programs, very specialized for specific purposes. Most often they are used to automates some simple process or operation, but there are examples of more complex applications. Number of malicious bots is not insignificant also. Such bots are used to create damage, denial of service, sort of enforcing some unwanted advertisement activity, posting some content or sending mass emails. In this paper we present one bot (bot – short of robot, well established name for internet robotic applications) which provides automatic search and process of business subjects in Republic of Serbia, regarding public available data on interactive web sites of APR (Agency for Business Subjects) and NBS (National Bank of Serbia). As illustration of potential of internet robotic applications, while concept is elevated to next level. This means that one bot can be used by some other software as tool. Meaning, software using other software as its own robot for his own purposes.

BOT GENESIS

Since all bots have specific usage, meaning their purpose is known in advance, first step in genesis of one internet bot is analysis of internet content/service with which will the bot interact instead of living person. In this example it is web portal of APR http://pretraga2.apr.gov.rs/ObjedinjenePretrage/Search/Search. Upon access of this webpage, user is required to choose type of search (by ID number of business subject or its name) and target cluster of business subjects to search trough. User is also required to enter the key term corresponding to chosen type of search.

After entering all required data in valid quantities/form, activating link "search", designated web page produces search result that may contain:

- Information that result of search is an empty set,
- Tabelar information with unique result,
- Table containing multiple results that met criteria of actual search.

In case that living person is doing this search, now looking at the results, he is required (in second or third case above) to click link „details" to make webpage produce and present available data about business subject. However, both for quantity and variety, data is grouped in 18 separate groups of data each consistent on different web page. So living person would must have to click each individual link to access all that data, and then to be able to visually spots, isolate and process wanted data. If some data needs to be stored for further use, living person must do it manually by marking the areas of webpage, then copying it into computer's clipboard, and pasting it to destination. Repetition of this operations for each individual data segment is mandatory by living user.

All this operations would be conducted much more effectively by bot instead of living person. Much faster, more efficient and without possibility of human-factor error. Such bot should have following functionalities:

- Ability to access APR website, presenting itself as regular living person using it's default web browser. Among other things, it must correctly accept, store and use all session variables and cookies[1][6], as any standard web browser would do.
- Ability to forward all identification data received on inital webpage, in addition to user-selectable data regarding the search parameters (target cluster of business subjects and key terms).
- Ability to analyze all received data from multiple result pages, select only relevant data in such form that it is usable for direct further processing.

Looking into structure of initial webpage for commencing the search, session variables, and form items relevant for request for search that need to compile in valid HTTP request[4] to make web server produce and present relevant wanted data.:

- rdbtnSelectInputType
- SearchByNameString
- SelectedRegisterId
- __RequestVerificationToken
- X-Requested-With
- JSESSIONID
- __utma
- __utmc
- __utmz
- SERVERID

Just first three elements noted are values that user is required to produce. Other elements are generated by webserver for particular session and presented on initial webpage. This elements serve to particular webserver to identify and track valid requests/queries. If any of this elements would be omitted or not contain valid data, such HTTP request would be rejected or ignored as invalid. Passing this stage, second part of bot's task is rather trivial. It simply consists of parsing received HTML content of result page, and selecting relevant data from it.

BOT IMPLEMENTATION

Having all parameters needed for proper bot functioning known and defined, one can advance to actual implementation of bot. Choice of tools to design, and software platform to execute bot depends of actual bot's nature. In our example target functionality of bot is to provide data in its raw form appropriate for transfer into some other software. Furthermore, it would be ideal if that other software can communicate directly with bot (commanding the bot), and being able to receive and process data resulting from bot's activity, without need for human intervention. In such case bot needs no user interface, since there is no human user which would require such interface. Instead, such bot needs software interface for communicating with other (commanding) software. This is possible to achieve in multiple manner. But in real life, most often such interface relations are implemented by following 2 manners:

1 Cookie - collection of data that websites often use to identify particular user

- Network client-server;
- Dynamic Link Library.

Both approaches have their advantages and floods. Clear and obvious advantage of network client-server bot interface is that it makes platform independent solution. It can be dislocated and independently execute away and separate form command software, on different computer, etc. One such implemented bot can effectively serve multiple command requests simultaneously. Main flood of this approach is the need for design of communication protocols which must be adopted and implemented by all programmers of command software, which requires additional resources, resulting in higher cost and slower development.

Clear advantage of second approach is to have bot implemented in dynamic link library (.dll file format in Microsoft OS family, or .so file format in Linux OS family), resulting in clean, simple and with zero dedicated development inclusion of bot functionality in preexisting software systems, with minimal effort and work hours of target software programmer. Besides that, need for extensive testing of communicating interface as in first approach is totally eliminated. Flood on the other hand is that such solution is not platform independent and must be separately compiled for every target platform separately. Also such approach requires that target library must be distributed with each copy of command software on each individual computer which run command software, or in case of upgrade/update of bot itself. This can happen in two reasons: Need for upgrade or bug fix the bot, or some change happened to structure of either request or result of bot's target webpage/webservice.

WEBSERVER'S SECURITY POLICIES AND BOT POLICIES

Many webservers and interactive webpages alongside exponential growth of internet, it's content and usage, implement special security mechanisms to protect itself from malicious or potentially harmful users or bots. Basic concept of such security is not having "content of interest for bots" available through static web links, but having "initial" webpage in between, which will throw some session elements and/or interactive elements that need to be included in HTTP request towards actual target webpage. Every standard web browser will pick up and deal properly with such elements by default, but as for bots, they need to be handled in program code, which eliminates most of "simple bots" that don't implement such capability. Beside that data, initial "barrier" can be topped up with additional security tokens generated at runtime. Such tokens can be generated in relation with client's IP address, actual date/time of request, cookies, etc.

Web Portal of APR from our example uses all this elements of "protection", as validation procedure that only living persons can access the data provided. However, using detailed analysis of generated target webpages and all identifying proper security elements, it is made possible to pass all that security countermeasures. After successful analysis of structure and content of initial webpage, all that is needed is to produce appropriate program code which will interact with targeted webpages.

Web server/services can also enforce bot protection based on denial of service to request coming from IP addresses verified as source of malicious activities. Such security policy is widely known as “black list policy”. Or to accept requests only from certain IP adddresses, which is known as “white list policy”.

Black list policy is mostly met when dealing with public web servers/services. On the other hand, web servers that primary serve narrow set of intranet clients, where no VPN infrastructure is available in various reasons, white list policy is often as measure of protection of access of unauthorized access.

Creators and users of bots fight the black list policy using anonymous proxy servers, which are easy accessible in waste amounts all over internet or Tor network[1], etc. There are also specific forms of black list policy, like for example limitation of usage only to user requests originating from specific geographic region/state/city, or denial of service based on same criteria. Originating of request can be determined by queering IP address of originating HTTP request against GeoIP database [2].

There are more advance techniques for protecting from bots. Most popular includes optically distorted images containing one or more words, thus demanding from user to enter word displayed on that image as means of verification that user is a living person and not a bot [3]. This technique is known as “capthca”. The idea behind this technique is that it is as easy for average living person to recognize the word displayed in distorted image, as it is hard for average bot at the same time to do so. Of course this technique is not foolproof, but it provides solid level of protection. There are also ways of passing or bypassing this technique, but it requires more efforts, resorts, etc. Generally speaking there are very few bots capable of this, because information that bot needs to extract or activity it needs to conduct need to be valuable enough to justify spending of such resources/efforts.

APR BOT

As said above, as an example of concept, in this paper is described how is designed and implemented one rather advanced bot which while simulating living person using standard web browser, access to APR’s web portal and requests a search by given term, receives and parses response got in return of request from webserver and produces results of relevant data in appropriate form. It all happens in one “user operation”, versus real living person interacting in multiple steps just to access data „scattered all over“.

Actual implementation of this bot is dual in its nature. One part is program code that does exactly operations described above, which is controlled by central code which as the same time implements TCP/IP server which interacts with third part of program code which is implemented as dynamic link library. This concept allows most universal usage of bot services, by numerous varieties of users simultaneously. Term “user” here refers to some pre-existing business software which can easily adopt the usage of documented functions from dynamic link library thus expanding and upgrading own functionality.

Regarding first functional part of the actual bot, following code (Listing 1.) displays core of program code which directly interacts with APR web portal.

```
URL url = new URL(aprURL);
HttpURLConnection conn= (HttpURLConnection) url.openConnection();
conn.setRequestMethod(“POST”);
conn.setDoOutput(true);
out=new DataOutputStream(conn.getOutputStream());
out.writeBytes(“”);
in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
while ((inputLine = in.readLine()) != null) {
 if (inputLine.indexOf(“__RequestVerificationToken”)>0){
 token=inputLine.substring(inputLine.indexOf(“value=”)+7),
 token=token.substring(0,token.indexOf(“\””));
 data = new HashMap<String, String>(),
 data.put(“__RequestVerificationToken”, token);
 data.put(“rdbtnSelectInputType”,”poslovnoIme”),
 data.put(“SearchByNameString”,keyword);
 data.put(“SelectedRegisterId”,””+repository);
data.put(“X-Requested-With”,”XMLHttpRequest”);
for (HttpCookie cookie: cookies);
data.put(“cookie”, cookie.getName() + “=” + cookie.getValue());
URL siteUrl=new URL(“http://pretraga2.apr.gov.rs/ObjedinjenePretrage/Search/SearchResult”);
conn=(HttpURLConnection)siteUrl.openConnection(conn.setRequestMethod(“POST”);
conn.setDoOutput(true);
out = new DataOutputStream(conn.getOutputStream());
Set<String> keys = data.keySet();
Iterator<String> keyIter = keys.iterator();
String req = “”;
for(int i=0; keyIter.hasNext(); i++) {
 Object key = keyIter.next();
 If (i!=0) req += “&”;
 req += key + “=” + URLEncoder.encode(data.get(key), “UTF-8”);
 }
out.writeBytes(req);
BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
String line = “”;
result=new LinkedList<String>();
while((line=in.readLine())!=null) result.add(line);
in.close();
```

Listing 1. Bot’s source code

Code is written in JAVA programming language, and on the listing only illustrative part of actual code is displayed. Actual code segment displayed accesses initial webpage and correctly indentifies, collects, stores and uses properly all security elements, adds user parameters to request and then sends such formed request which the webserver will accept and process as if living user made the request using regular web browser. Code that follows is trivial part just parsing the HTML reply from webserver.

```
function connect(uname:PChar; pass:PChar)
function searchByKeyWord(pojam:PChar)
function isResultEmpty:boolean
function gotoNextResult:boolean
function getResultName:PChar
function fetchResultDetails
function getSkrIme:PChar
function getMatBr:PChar
function getPravnaForma:PChar
function getOpstina:PChar
function getMesto:PChar
function getAdresa:PChar
function getDatumOsn:PChar
function getPIB:PChar
function getSFDel:Pchar
function getNextTR:PChar
function getZakZastIme:PChar
function getZakZastJMBG:PChar
function getZakZastFunkc:PChar
```

Listing 2. List of some functions exported in library

At listing above (Listing 2.) are displayed some of exported (public) functions in dynamic link library which represent universal software interface for using APR bot's services.

So software using this bot's services would do it in a following manner:

1. Call to function connect passing it assigned username and password.
2. Call to function searchByKeyWord passing it key term (part or whole) name of business subject of interest.
3. Call to function isResultEmpty, and by interpreting its return value user-software can tell if APR database contains one, more or none business subject that meets search criteria.
4. Call to function gotoNextResult results with positioning the cursor of internal list of results on next result of search.
5. Call to function getResultName results with whole official individual name of individual business subject within search result set at which record internal cursor is positioned at the moment.
6. If current element in result set is the only element in result set, then call to function fetchResultDetails is next step. The function itself will make multiple simultaneous queries on APR's database and extract data in all 17 sections in "one expense" regarding the time consumption. Instead of making 17 sequential queries, using and launching 17 separate threads in the same time, saves considerable amount of time.
7. Afterwards, relevant data can be acquired from bot's memory by calling appropriate function like getPunoIme, or getPIB, getOpstina, etc.
8. If, in the other hand, result of initial search returned result set containing multiple items, by choosing specific item from result set by for example calling function positionToResultName and passing given business subject name as parameter, bot sets in ready state to continue providing further detailed data as mentioned in steps 6. and 7.

MORE ADVANCED POSSIBILITIES

So far bot was presented as direct tool that would search internet content in most efficient way and present the appropriate data on demand. Beside this direct usage "on demand", this bot can be used in another manner.

For example let's consider some business subject that is registered in Republic of Serbia, doing business with other business subjects in republic of Serbia. During past several years, due to global economic crisis, some business subjects get in state of financial insolvencies, which results in irregular payments of its business obligations, both towards the State or other business subjects. This further leads to blockage of business bank accounts either by the State or some other business subject. This blockage is noted in central database of NBS and that data is public available too.

So let's consider some large scale business subject that does business with numerous other business subjects in Republic of Serbia. In order to protect its business knowing such information about other business subjects that they do business with on regular basis, can be very valuable. To monitor regullary status of particular business subjects falling into blockage or exiting from it, amount of blockage and history of such state of potential new customer. Doing do manually would require significant human resources and massive work hours of one or more employees in charge to for that. Not to mention that living person is prone to honest mistakes of mistyping, overlooking etc.

It would be most valuable to be able to just make a "list of interest", in business software which will using above described bot inquire official online databases and compare previous data looking for any relevant change, for example on daily basis, and report to appointed staff in company via email for example, notifying them of relevant changes of status of some business subject from "list of interest".

PERFORMANCE OF EXPERIMENTAL BOT

While doing this research, in parallel, Android[2] application is developed that utilizes particular bot. It's called

2 Android –OS based on Linux kernel, primarily intended for mobile handheld devices

"Privredni detektiv" and it provide easy and comfortable way of searching and displaying relevant data of business subjects registered in Republic of Serbia. Its sole purpose is purely educational; it can be downloaded for free from following link:

ftp://entercom.digipro.rs/Public/androidapps/PrivDetetktiv.apk

Key benefits of using this application are:

Simplicity of use

Efficiency

Presenting verified data fetched from different sources

Simplicity is amplified in a manner that user need not to choose specific register of business subject he needs info about. Bot will search all registers for user, and display combined results. And all the data from different sources will be collected, filtered and presented in appropriate form by just "one user click", instead of multiple searches which would produce partial data, which need to be stored and combined by user manually

Efficiency is so high, that all above mentioned searches will not take place sequentially, but simultaneously using Threads[3]. To be more precise, for each specific user operation (either search of specific register, or clicking on partial detail section of chosen business subject), bot "launches" separate thread, which all execute roughly at same time thus reducing sole time of execution by 17 times at minimum (that many user clicks&views are needed to access all the data). In average efficiency rises more than 20 times just in sole time of execution of search and all detailed data fetching.

Time savings in storing and collecting all the data which user would have to do manually is obviously even much greater.

As mentioned before, example bot doesn't access only APR database, but NBS database too. After having unique ID from APR of chosen business subject, it launches separate thread to access NBS's database and fetch data about bank accounts, etc.

Then all fetched data are sorted and displayed in favorable manner.

3 Thread- Functional part of computer code that can run concurrently with more other threads at the same time inside one main thread.

CONCLUSION

Potential and scope of applications for internet robotic application are very high and broad. Especially for online content or webserver/webservices that doesn't offer usable API for their services, which is segment of internet programming with highest growth rate in past few years. As mainstream of computer software migrates from desktop and classic client-server systems towards cloud computing, this trend will continue.

Weather internet robotic applications are just temporary phenomenon, or will grow in numbers, future will tell. But in general, we can fairly assume that as long are man-made operations on internet, as in "physical realm", robots that would do man's work are always welcome and can be put to good use.

ACKNOWLEDGMENT

This project is supported by Ministry of Science and Technological Development of Republic of Serbia, trough ongoing projects TR32054 and ON174008

REFERENCES

[1] V. Stanojević, „Anonimnost na internetu –Tor rešenje", master rad, Univerzitet Singidunum 2013.

[2] Jafari, S.J., Naji, H. "GeoIP clustering", Information and Knowledge Technology (IKT) Conference, 2013

[3] Jing-Song Cui, Jing-Ting Mei, Wu-Zhou Zhang, Xia Wang, Da Zhang „A CAPTCHA Implementation Based on Moving Objects Recognition Problem", International Conference on E-Business and E-Government 2010.

[4] Sosinsky, „Networking Bible", Wiley Publishing, 2009.

[5] S. Davidoff, J. Ham, „Network Forensics", Prentice hall 2012

[6] James F. Kurose, Keith W. Ross,"Computer Networking: A Top-Down Approach" ,3th Edition, 20