REST API and Http Basic Authentication

Programming Real-time NLP in WEX 11.0.1
Contents
Introduction .................................................................................................................................................. 4
Programmatic Methods for Supplying BasicAuth parameters. ................................................................. 6
  Browser Workaround.................................................................................................................................... 7
  cURL and wget Workaround ..................................................................................................................... 7
  Java Code Workaround .............................................................................................................................. 8
Appendix A: suppress the Confirmation dialog in Firefox ........................................................................ 11
<table>
<thead>
<tr>
<th>Author</th>
<th>Date</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kameron Cole</td>
<td>July 18, 2016</td>
<td>0.1</td>
</tr>
</tbody>
</table>

kameroncole@us.ibm.com
Introduction
This is a brief paper on coding using the REST API, focused on the interaction of REST, which is an HTTP-centric framework, and HTTP Basic Authentication.

In the case of WEX, it is possible to enable only HTTP Basic Authentication:

Most users will expect to have to log in to the various web applications, like the Enterprise Search and the Miner applications, but most are surprised when the issue a REST query and see this:
It is critical to point out here that this dialog does NOT come from the WEX product, nor is it related to the WEX REST API. As mentioned above, this is a requirement for browsers and application servers per the HTTP Specification:

https://en.wikipedia.org/wiki/Basic_access_authentication

**NOTE: Do not confuse authentication, with authorization. Indeed, the WEX product APIs are required to provide parameters for authorization – but not authentication. For example, the REST Admin API must be authorized using parameters api_username=<username>&api_password=<password>**
REST Admin Examples

Get the build info -

Programmatic Methods for Supplying BasicAuth parameters.

While IBM does not offer support for HTTP Basic Authorization programming, it is not difficult to find several means by which REST developers can “get around” the BasicAuthentication. However, it is critical to note that it is NOT ADVISED by IBM nor probably any server administrator to “get past” a security mechanism, there may be cases when for the sake of a Proof of Concept, a developer would want to use means to work around the bounds of the BasicAuthentication, for programmatic access.

NOTE: The preferred means of securing all http traffic for IBM is the use of the LTPA token, and methods of Single Sign On. If LTPA has been configured, HTTP Basic Authentication is surpassed by the presence of LTPA2 tokens in the HTTP Header. Developers should familiarize themselves with IBM’s LTPA technology:
**Browser Workaround**

If you want to enter the Basic Authentication in the browser, along with your REST query, simply use the format:

http://<username>:<password>@www.example.com/webcallback?foo=bar

For example,

As seen in the image above, some browsers will add a confirmation pop-up. Read Appendix A below for the Firefox workaround for this. Once you fix this, you will no longer be challenged at the browser.

**cURL and wget Workaround**

There are some excellent IBM articles about using these tools specifically with REST coding:

http://alvinalexander.com/web/using-curl-scripts-to-test-restful-web-services


While these articles go into great depth, the problem at hand is solved quite simply:

1. cURL : curl -user username:password URL.
2. wget, you can use --ask-password option.
Java Code Workaround

Here is a simple Java example:

```java
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import org.apache.commons.codec.binary.Base64;

public class ConnectToUrlUsingBasicAuthentication {
    public static void main(String[] args) {
        try {
            String name = "kameroncole@us.ibm.com";
            String password = "XXXXX";

            String authString = name +":"+ password;
            System.out.println("auth string: " + authString);
            byte[] authEncBytes = Base64.encodeBase64(authString.getBytes());
            String authStringEnc = new String(authEncBytes);
            System.out.println("Base64 encoded auth string: " + authStringEnc);

            URL url = new URL(webPage);
            URLConnection urlConnection = url.openConnection();
            urlConnection.setRequestProperty("Authorization", "Basic " + authStringEnc);
            InputStream is = urlConnection.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);

            int numCharsRead;
            char[] charArray = new char[1024];
            StringBuffer sb = new StringBuffer();
            while ((numCharsRead = isr.read(charArray)) > 0) {
                sb.append(charArray, 0, numCharsRead);
            }
            String result = sb.toString();
            System.out.println("*** BEGIN ***");
            System.out.println(result);
            System.out.println("*** END ***");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Here is the output:

```
[kameroncole@icahost httpbasicauth]$ java -cp .:/lib/commons-codec-1.10 commons-codec-1.10.jar
ConnectToUrlUsingBasicAuthentication
auth string: kameroncole@us.ibm.com:XXXXX
Base64 encoded auth string: a2FtZXJvbmNvbGVBVAdXMaWJtLmNvbTpmdTQzZ2ZreQ==
```
<?xml version='1.0' encoding='UTF-8'?>


  <atom:title>Search results for query 'chain' on collection 61944</atom:title>


  <atom:author>
    <atom:name>Discovery Search REST API</atom:name>
  </atom:author>


  <es:evaluationTruncation>false</es:evaluationTruncation>


  <opensearch:totalResults>1</opensearch:totalResults>

  <es:totalResultsType>estimated</es:totalResultsType>

  <es:numberOfAvailableResults>1</es:numberOfAvailableResults>

  <es:numberOfEstimatedResults>1</es:numberOfEstimatedResults>

  <opensearch:Query role="request" searchTerms="chain"/>

  <es:query role="full" searchTerms="chain"/>

  <es:resultGroups>
    <es:resultGroup id="Y29tLm1ibS51cy5zZWFyY2gucmVzdWx0Z3JvdXAuQ2F0Y2hBbGw==" label="CATCH ALL" href="http://icahost.ibm.localdomain:8393/api/v10/search?collection=61944&resultGroup=Y29tLm1ibS51cy5zZWFyY2gucmVzdWx0Z3JvdXAuQ2F0Y2hBbGw%3D" query="chain"/>
  </es:resultGroups>

  <opensearch:startIndex>0</opensearch:startIndex>

</feed>
Appendix A: suppress the Confirmation dialog in Firefox
https://bugzilla.mozilla.org/show_bug.cgi?id=243572

You can effectively disable this "feature" by adjusting this hidden user preference:

    network.http.phishy-userpass-length

and set the numeric value to some large number.

Just load about:config, right click and choose New->Integer.
Enter this pref name and set the numeric value to some large number.

NOTE: This solution from Mozilla Support does not work in all editions. Here is the workaround:

You must add this line to the user.js file in the user’s Profile Directory:

```javascript
user_pref("network.http.phishy-userpass-length", 1024);
```

You will need to create user.js, and find your Profile directory:
Troubleshooting Information

This page contains technical information that might be useful when you're trying to solve a problem. If you are looking for answers to common questions about Firefox, check out our support website.

---

### Application Basics

<table>
<thead>
<tr>
<th>Name</th>
<th>Firefox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>42.0</td>
</tr>
<tr>
<td>Build ID</td>
<td>2015032151421</td>
</tr>
<tr>
<td>Update History</td>
<td>Show Update History</td>
</tr>
<tr>
<td>Update Channel</td>
<td>release</td>
</tr>
<tr>
<td>User Agent</td>
<td>Mozilla/5.0 (X11; Linux x86_64; rv:42.0) Gecko/20100101 Firefox/42.0</td>
</tr>
<tr>
<td>Profile Directory</td>
<td>Open Directory</td>
</tr>
<tr>
<td>Enabled Plugins</td>
<td>about:plugins</td>
</tr>
</tbody>
</table>

---

```bash
user_pref("network.http.phishy-userpass-length",1824);
```

---

-- INSERT --