Create search widgets to proxied resources

The /form URL allows you to place search forms on public web pages. It is similar to /login, except it expects to receive additional form variables which need to be sent to the remote database. When using the /form URL, the requests of local users are sent directly on to the remote database unproxied, whereas unauthenticated remote users are required to log in first, then their requests are sent on proxied to the database.

Note: This method only works with remote search pages that support the get method for form processing, not the post method.

This document describes how to construct such a form for use with Google Scholar. It assumes that Google Scholar has already been used to configure EZproxy for use with Google Scholar.

An example that does not work

At first glance, it might seem that the example below would allow you to place a search form on a public page and it would operate in the manner described in the overview.

```
  <input maxlength="256" size="40" name="q" value="">
  <input type="submit" value="Search">
</form>
```

However, the EZproxy /login URL makes very specific assumptions about the information it will receive, and receiving arbitrary form variables such as "q" in this form is not one of these assumptions, causing that variable to be discarded and preventing this form from working as desired.

An example that does work

```
<form method="post" action="http://ezproxy.yourlib.org:2048/form?qurl=http%3a%2f%2fscholar.google.com%2fscholar">
  <input maxlength="256" size="40" name="q" value="">
  <input type="submit" value="Search">
</form>
```

The HTML form tag in this new version has three key differences.

1. The method has been changed from get to post. /form will only work correctly if you specify the post method. Even though you specify post here, when the request is finally sent to the remote server, it will always be sent to that server using the get method.

2. In action, /login has been replaced by /form.

3. The destination URL is no longer specified using url= but instead is specified as qurl= using qurl= in place of url= and specifying the destination...
URL with URL escaped encoding. To apply URL escaped encoding to an arbitrary URL, EZproxy provides a /encodeurl URL that you can access with a URL similar to:

```
http://ezproxy.yourlib.org:2048/encodeurl
```

On the /encodeurl page, you provide the original URL, press enter, then you receive the escaped version to use following qurl=:

Once you place this version of the form on a public page, both your local and remote users should be able to use it to perform searches, with local users required to authenticate but then proxied through to perform the search.