Learn how to use the AnonymousURL config.txt directive to allow users to access select content in EZproxy without authenticating.

**AnonymousURL** allows users to access select content through your EZproxy server without authenticating. **AnonymousURL** is a position-dependent directive that interacts with database stanzas.

Caution: This directive directs EZproxy to proxy specific URL patterns without the request possessing an EZproxy cookie. If you are not clear about how this directive works, OCLC encourages you not to use it. If you need help constructing such entries, contact [OCLC Support](https://help.oclc.org/Library_Management/EZproxy/Configure_resources/AnonymousURL) for assistance.

**Qualifiers**

<table>
<thead>
<tr>
<th>QUALIFIER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Options</td>
<td>Specifies that the rest of the directive should match for OPTIONS method requests from the browser but not others such as GET and POST.</td>
</tr>
</tbody>
</table>

**Syntax**

EZproxy V6.1 added the **-Options** qualifier to the **AnonymousURL** directive, making the new complete syntax:

```bash
AnonymousURL -RE -CS -Options +|-- wildurl
```

The line should appear at the beginning of the database stanza it should impact. Suggested usage includes the following line as the final line of any stanza with an **AnonymousURL** directive to limit the directive's behavior to the intended stanza:

```bash
AnonymousURL -*
```

**Examples**

The following examples provide some use cases where the **AnonymousURL** directive would be appropriate. RSS readers If you use an RSS reader, the reader needs to be able to request the XML file through EZproxy so the links will be rewritten, but the RSS reader itself runs at independent times outside the context of EZproxy. In most instances, the RSS feeds themselves are designed for public use by other RSS readers, so the content does not need to be secured from access by EZproxy. Having EZproxy allow the requests through without an EZproxy cookie resolves this.

**RSS feeds example**

If the fictional Research Database provides RSS feeds at http://www.researchdb.com/rss/feename.xml, you might use the following stanza to allow access to these feeds.
You would then need to determine the EZproxy version of the hostname for www.somedb.com and use it to construct a URL such as:


**External streaming media players example**

When playing media, especially streaming media, the browser may launch an external program to deliver the content. When the URL reaches that external program, it does not possess EZproxy's session cookie, so the request will be blocked. In these scenarios, vendors have accepted the necessity of allowing the portions of the URLs relating to that media content to proxy without EZproxy actually being able to know for certain that the remote user has authenticated and is allowed to access this resource.

If the fictional News Database uses an external program to deliver streaming WMV files, the following database definition could be necessary for users to access that content:

```plaintext
AnonymousURL +*.wmv
Title News Database
URL http://www.newsdb.com
DJ newsdb.com
AnonymousURL -*
```

**Java applets example**

Some Java applets don't have access to send cookies. Some database stanzas will contain an AnonymousURL directive to allow processing of the content presented with these applets.

If the fictional Art Database uses Java applets that cannot send cookies, you might need to construct the database stanzas as follows to allow access to JPGs and GIFs:

```plaintext
AnonymousURL +*.gif
AnonymousURL +*.jpg
Title Art Database
URL http://www.artdatabase.com
DJ artdb.com
AnonymousURL -*
```

**Ajax requests example**

A new use case occurs when making Ajax requests to a different server. When this happens, an OPTIONS request may be made without cookies present. If EZproxy does not proxy this, the ability to access the remote content is blocked.

**OPTIONS requests with a CORS request example**

When browsers send an OPTIONS request, such as when preflighting a CORS (cross-origin resource
sharing)request, they do not send along their cookies as part of the request. When EZproxy receives the request with its cookie, it can't confirm that the user is authenticated, so it creates a redirect to authenticate. Using the -OPTIONS qualifier with AnonymousURL allows EZproxy to handle cases like this.

For example, if EZproxy receives an OPTIONS request from the fictional Research Database, the following database stanza could be used to tell EZproxy that the OPTIONS request can be processed without authentication, but any normal request must have the authentication cookie present:

<table>
<thead>
<tr>
<th>AnonymousURL -OPTIONS +<a href="http://api.researchdb.com">http://api.researchdb.com</a>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: Research Database</td>
</tr>
<tr>
<td>URL: <a href="http://researchdb.com">http://researchdb.com</a></td>
</tr>
<tr>
<td>DJ: researchdb.com</td>
</tr>
<tr>
<td>AnonymousURL -*</td>
</tr>
</tbody>
</table>

https://help.oclc.org/Library_Management/EZproxy/Configure_resources/AnonymousURL
Printed: Mon, 06 Apr 2020 05:32:18 GMT