Use HTTP caching to improve performance

As a VIVO implementation grows in size and tracks more and more scholarly activity, profile pages can be pulling in hundreds of relationships to render the page, which results in more data being retrieved from the underlying triple store and longer page load times. For example, a profile page for a faculty member with hundreds of publications, which isn't uncommon, can lead to multiple second page loads.

Instead of querying the database each time a page is loaded, a cached version of the page can be served, provided the user is not logged in. VIVO supports HTTP caching directly. To enable, uncomment the "http.createCacheHeaders = true" line in runtime.properties:

runtime.properties

```
# Tell VIVO to generate HTTP headers on its responses to facilitate caching the
# profile pages that it creates.
#
# For more information, see this wiki page:
# https://wiki.duraspace.org/display/VIVO/Use+HTTP+caching+to+improve+performance
#
# Developers will likely want to leave caching disabled, since a change to a
# Freemarker template or to a Java class would not cause the page to be
# considered stale.
#
  http.createCacheHeaders = true
```

VIVO will now generate eTags for caching, which are stored in VIVO's Solr index. More information is available from Ted Lawless, who originally demonstrated the eTag method, here.

Next, enable mod_cache in Apache by uncommenting LoadModule lines in httpd.conf:

httpd.conf

```
LoadModule cache_module modules/mod_cache.so
LoadModule cache_disk_module modules/mod_cache_disk.so
```

and adding the following configuration lines to httpd.conf or in its own .conf file within Apache's conf.d directory:

mod_cache.conf

```
#The default expire needs to be 0 in a self-editing environment so that E-Tags can be reverified.
#Requests to cached URLs that haven't expired will never reach the VIVO web application.
#

<IfModule mod_cache.c>
     CacheRoot /var/cache/apache2
     CacheEnable disk /display
     CacheEnable disk /individual
     CacheIgnoreNoLastMod On
     CacheDefaultExpire 0
     CacheMaxExpire 0
     CacheIgnoreHeaders Set-Cookie
</IfModule>
```

The above configuration was provided by Ted Lawless. Restart Apache and Tomcat. Large pages should now load significantly faster for logged-out users.

You can verify http caching is occurring by looking in the directory specified as CacheRoot and seeing if files are being added. You can also use your browser's debugging tools, like Firebug or Chrome debug tools, to inspect the HTTP status code of the response for a profile page. In Chrome, enable Developer Tools (View > Developer > Developer Tools, or I) and select 'Network' on the pane that appears. Cached pages will return a 304 "Not Modified" response.