VIVO Programmer's Notes

- Architecture
- Diagnosing problems in VIVO
- The Developer Panel
- VIVO API

Topics for VIVO developers, or advanced VIVO customizers.

Explanations of particular parts of the VIVO architecture and implementation.

Architecture

Some insights into the structure of the VIVO/Vitro code base.

- HTTP Cache Awareness — VIVO adds headers to some HTTP responses, to assist in caching profile pages. Since release 1.6.
- Interface to external authentication systems — More details on how VIVO will integrate with a Single Sign-On system like Shibboleth or CUWebAuth.
- Software Architecture Overview — A diagram of the VIVO software architecture, with notes.
- VIVO and the Solr search engine — VIVO and Solr are two distinct web applications that act as one. Solr gives users the ability to search the VIVO data. VIVO also uses Solr for some of its internal data retrieval.

Diagnosing problems in VIVO

Here are some techniques to use when VIVO is not behaving as it should. Diagnose the problem, isolate it, and find a solution.

- Troubleshooting VIVO Search — If you believe that VIVO contains data, but you can't find any, you may very well have a problem with Solr.

The Developer Panel

Diagnostic tools that can help you figure out what VIVO is doing.

VIVO API

How to get data into or out of a running VIVO instance.

- Linked Open Data - requests and responses — All of the public RDF in VIVO is available in response to Linked Open Data requests. VIVO responds to standard LOD requests, and also some request formats that are particular to VIVO.
- The Search Indexing service — The Search Indexing service allows you to request updates to the VIVO search index, but only for a specific set of URIs. If you know which search records are out of date, this is faster than rebuilding the entire index.
- The SPARQL Update API — Remote applications can perform SPARQL Update calls to add RDF to VIVO, or to remove existing RDF. Since VIVO 1.6.