Documentation for Prospero

Note: All directories and files mentioned below are with reference to the server prospero.netherweb.com.

These files are also available via the Prospero project on sourceforge, https://sourceforge.net/projects/prospero .

/Users/prospero/prospero/modules/update_modules.rb

This is the code which loops through all the modules in the modules directory. For each module, it checks validity and saves it into the database.

/Users/prospero/prospero/src/control/server.rb

First of all, let's keep in mind that there are different kinds of servers at play in Prospero.

These are: the XML RPC server and the module server.

The XML RPC server runs on port 9090 and defines RPC calls.

Modules can define their own RPC calls.

/Users/prospero/prospero/modules/<modulename>

where <modulename> is the name of a module, e.g.:

/Users/prospero/prospero/modules/facebook

/Users/prospero/prospero/modules/<modulename>/html

Loads custom functions for the module

Defines base RPC calls

Loads HTML modules

/Users/prospero/prospero/src/control/start_collectors.rb

Loads the collector (where applicable) for each module.

Creates an instance of the updater class

/Users/prospero/prospero/src/control/updater.rb

The updater class populates the user table with data. The way it does this is as follows:

All existing instances of the User class (defined in ../data/db.rb) are read in, in a loop.

For each existing user instance, run the collector (via the “populate” method, defined in /Users/prospero/prospero/src/control/<modulename>_collector.rb), [need to check with Ben if this is the right collector method] saving the results in the database.

Collectors: /Users/prospero/prospero/src/control/<modulename>_collector.rb

We need to have collectors because we want to cache {key,value} pairs. For example, for the Facebook collector, we want names of users to be keys and data that we are gathering from their Facebook record (for example, their picture, their favorite movie, etc.) as values.

The collector class, defined in the above file, has the name <Modulename>Collector. Two important methods are defined in its API: 'name' and 'get_values'. The method 'name' gets the collector name in lower case – typically, '<modulename>collect.' and the method 'get_values' gets the collected data.