Architecture Sample Work

* Service based architecture
* One service per task:
  1. Upload
  2. Unzip
  3. Build with javac
  4. Build with maven
  5. Rezip
  6. Download
  7. Login
  8. Logout
  9. Copy between services
* Services implemented either as a HTTPExchange wrapper around an operating system script, subclassing RuntimeService.
* Or as a subclass of Service ( parent of RuntimeService ), with specific tasks implemented in Java, instead of operating system script.
* Two groups:
  1. Runtime Services:
     + Unzip
     + Build with javac
     + Build with Maven
     + Rezip
     + Download
     + Copy between services
  2. Services
     + Login
     + Logout
     + Upload
     + Download
* Very very simple code, everything written from the ground up.
* Sessions ? Yes.
* Clients ? Yes.
* Object store ? Yes ( see cavedb, and inspect the database under the cave directory ).
* HTML templating ? Yes ( see the HTML pages in jar.io/pages )
* Reason for the ground up approach was to emphasise architectural choices of the author, rather than choices imposed or suggested by a Framework. Also provides total flexibility.

Improvements or Enhancements

* Refactor code ( particularly in Service class )
* System can scale out to run services on separate machines by making copyservice copy between machines, and having a stand alone Cavedb object store service.