FirstReport

Google Summer of Code 2014 Report 1

Project Name: NUTCH-841 Create a Wicket-based Web Application for Nutch 2.X

Report date: 19th June 2014

Student Name: Fjodor Vershinin

Mentor Name: Lewis John McGibbney (lewismc)

Development CodeBase: https://bitbucket.org/feodorv/uinutch/

Project description

Main goal of this project is to create an Apache Wicket-based Web GUI for Apache Nutch 2.X.

Review of Previous Actions

No actions are available as this is the first report.

Objectives

Explore Nutch Documentation

Firstly, I decided to make some background research about Apache Nutch in Russian segment of Internet. Then I briefly reviewed Nutch wiki, but especially paid attention to GUI specification page and Nutch tutorials.

Workspace Setup

Nutch compilation process is built with Ant+Ivy. I'd previously experience with only Maven, so I've spend some time on workspace setup. Also, I prefer to use Mercurial because of very continent MQ plugin, which I really missing in SVN and GIT. Fortunately, Mercurial has also SVN/GIT bridges, so it was not that troublesome to configure it. As a benefit I got very convenient tool to work with patch queues.

Contributions to Nutch community

- NUTCH-1731 I'd fixed NPE bug, added ability to stop remote server, and re-factored code so it uses Apache commons-cli library for command line parsing and automatic help message generation.

Some experimental crawling

Then I decided to make some test crawling sessions, in order to get into Nutch mechanics and plugin configuration. Despite very verbose wiki documentation it took some time to start, because of Hbase, Elasticsearch, and Hadoop setup. Using this software stack I'd crawled some web pages and got some Nutch user experience.

Review previous GUI application

I'd checkout legacy GUI application from Github. Despite that I'd spend some time to get it running, it was quite useful, because I get information about most important features which should be implemented in the first row. From my point of view, most important features were:

1. Seed file upload
2. Component which gives ability to change nutch default settings.

3. Search in crawling results
4. Crawling statistics
5. Authentication
6. Instances management
7. Plugins support

Create HTML prototype

I created HTML prototype to make some experiments with Twitter Bootstrap and as a final result import legacy application HTML pages into my project. Wicket gives ability to use such prototype pages without any problem just by adding wicket:id properties.HTML prototype is far from complete, but you can get latest state by this link https://bitbucket.org/feodorv/prototype/get/default.tar.gz or clone mercurial repo using "hg clone https://feodorv@bitbucket.org/feodorv/prototype"

Nutch REST API research

My first serious meeting with Nutch code base. I decided that it will be good entry point if I start with REST API. Then I can better understand, what features have been already implemented on server side, which drawbacks there exists and how application works. It was quite useful despite it took quite a lot of time because of code complexity and cohesion. After this step I'd got better experience with Nutch internal architecture and source code.

Refactoring of NUTCH Rest API

After reviewing of this API I realized, that code requires refactoring. Most of this code has been written two years ago with plain old Restlet library. We decided that I can use JAX-RS interface, because it is standard for modern REST services. Restlet has been left as back-end, but it can be replaced now without any effort. Also, refactoring has decreased code cohesion and complexity. These changes were proposed as patch in NUTCH-1769 issue. Also, I would provide unit/integration tests for REST API, but it is out of GSoC scope.

Implement application skeleton

Most time consuming part. I'd spend some time on exploring technologies, such as embedded Jetty, Wicket Spring integration and so on. Then I asked for some help from professional Java web-developer, in order to get some advice how to setup initial application skeleton. He gave me some Wicket samples from his projects and routed me to the right documentation about Wicket and Spring. Implemented features in application skeleton:

- RESTful client to Nutch API
- GUI application configuration component
- Base page structures
- Navigation
- Twitter Bootstrap support
- Unit tests
- Integration tests
- Plugin support

You can clone mercurial repository using "hg clone https://feodorv@bitbucket.org/feodorv/uinutch" command or download latest version by this link https://bitbucket.org/feodorv/uinutch/get/default.tar.gz

Future Actions

- Add ability to get logs by REST API
- Implement generic crawl cycle in GUI
- Add ability upload seed files (or post seed data) by REST API

Mentors Comments

It has been some three weeks now since I am aware that Fjodor actually began coding on this project. He stated that he would be delaying the start due to exam committments and I was fine with this. My main concern was what would be the tangible outcome for mid-term reporting based upon how quickly he could get upt-to-speed with the rather complex nature of Nutch 2.X. As far as I am concerned, and based on his initial report as above, I am pretty satisfied that he understands the high level view of Nutch 2.X including the layers of Nutch as the Hadoop-based application, Gora as the storage abstraction later, HBase (amongst others) as the WebPage/Host storage mechanism, Solr as the indexing server and Hadoop as the framework for running Nutch jobs on. This in itself takes most people much longer than 3 weeks so I am reasonablty happy with his progress. I would like Fjodor to be contributing his documentation at regular intervals to **THIS WIKI**, this serves a number of purposes

- It means I can track his progress between reporting sessions
- It means we have one place where all of the documentation resides
- It means we can direct GSoC admins, etc here when reporting comes around
- It means that future students and/or anyone else can find how the project went
- ... etc

I would like to see Fjodor take action on the following items

- ~~If possible create a graphic of the REST API as it exists in his proposed patch for NUTCH-1769 API refactoring this should only include the information included in his above commentary on the topic.~~
- ~~Provide links to the **HTML Prototype**, I have not seen any of this code and therefore cannot assert that progress has been made as described above.~~
- ~~Provide links/patches for the **application skeleton** as stated above... I have yet to see any code.~~

All in all, I am happy that Fjodor is being resourceful e.g. approaching Java developers to get help, etc. Fjodor has been a quiet student and has not asked a lot of me so far, if he wishes to change this moving forward then I am very willing to engage more with him in his on going work.

Signed: Lewis John McGibbney (lewismc)