# NutchAdministrationUserInterface

## Proposal Nutch appliance / Nutch admin gui

### Summary:

The goal is to extend Apach Nutch with a comfortable web based administration user interface to monitor, configure and manage one or a set of Nutch search system instances through the REST-API. This will tie together a number of issues, ultimately resulting in a Nutch 2.0 Webapp



### Vision:

Have a distribution of nutch that only needs to be decompressed and can be started in three different modi.

- Starting nutch in a single master mode - starts a local map reduce nutch instance by launching a simple nutch admin gui.
- Starting nutch in a multi master mode - starts nutch jobtracker, namenode and admin gui.
- Starting nutch in a worker mode - starts data node and trasktracker.

Command line administration and configuration as it is used today should be still available. The described admin gui can never replace a fine tuned, shell script administered nutch, but it can help to get users faster and easier started with nutch. A easy to use webbased configuration and administration gui would also help to increase the user basis of nutch.

### Use cases:

Running nutch as a web master with minimum efforts for installation and integration. Configuration and iterative crawl scheduling can be configured in the admin gui. Also the open search servlet can run in the same embed web container as the admin gui.

Decompress starts one nutch application, but runs different search systems from the same code basis and jvm that have completely different configuration spaces. This could be interesting to run a web page, extranet and intranet search system where different parsers, threads and scheduling times should be configured. It would be possible to store the data's of the nutch instance in different folders, so three different web contexts can access these data independently. It would be also possible to run the search backend on one server, but running the search ui itself on a other system as long as the other system can access the data.

Running a horizontal special interest search engine where data fits not to one computer hdd and cpu performance anymore but make the configuration and management still end user friendly.

### Look and feel Admin Gui:

This link provides the best working prototype of an example admin gui, it also provides a heap of material relating to what kind and level of functionality the Nutch webapp should support.

#### A New Nutch Instance

`instanceNew.jpg`

#### Congiguration UI

`configuration.jpg`

### URL Upload

`urlUpload.jpg`

### Example Crawl 1

`crawl1.jpg`

### Example Crawl 2

`crawl2.jpg`

### Example Crawl 3

`crawl3.jpg`

## Description Admin Gui:

There are three main functionalities of the admin gui

### Monitoring:

The tab "system status" gives the status of all connected tasktrackers and also data nodes and there capabilities. Under "Tasks & Jobs" a job process overview as the Jobtracker info server provides today will be available. Also the job detail page will pretty much show the same information as it does today, may just some more meta information like on which tasktracker a specific tasks runs actually.

### Configuration:

Behind the tab "Configuration" all properties of a nutch instance can be configured. We hope we can split the today large number of properties somehow in logically pieces. The tab "Scheduling" hosts a screen to configure scheduling based job submission. Therefore we plan to add a scheduling mechanism like quartz into the admin gui. Also "Exclude URLs" is somehow part of the configuration section.

### Management:

The management console is available behind the tab " Management". Here a user can create a new segment and manually starts the processing steps of a segment life-cycle like fetching, update crawl db, index and add the segment index to the actually searchable indexes. Beside this manually maintenance of segment life cycle steps it is possible to configure scheduler based jobs that runs the complete life-cycle process for the user automatically.

## Technically description of the admin gui:

The admin gui will be a web application running in a embedded servlet container (jetty) as the job tracker status page already does today.

The admin gui will provide an extension point, that allows to add more functionality to the admin gui by providing a jsp snippet for ui integration and a set of java classes that have to implement the extension point itself. All installed extensions will be shown in tabs and can be used to configure locally running custom plugins also as remote resources used within nutch. Since it will be able to have several nutch instances using the same code base there will be multiple working directories for each instance. ($Nutch/intranetInstance or $Nutch/WebpageInstance) Beside a segment folder, crawl and link database a working directory will also contain instance specific configuration files like nutch-site.xml and mapred-default.xml, in short all instance related data will be stored in the instance folder.

All configurations will be stored as files in an instance folder, but will be also hold in memory to use as non static nutchConf to pass to tools calls. Nutch tools that are now started by a a shell script will be started by using an API call where we pass the nutchConf instance, this require that all tools have such an API. We should discuss if we run the tools in an extra thread and catch all throwables or if the tools are started in a separated jvm as the tasktracker does today. I personal prefer the first solution.

To track the status of a segment we can write small text files into the segment as the index in nutch version 0.7 does already. To identify if a segment is already indexed we can store a index.done file in the segment folder. Having this instance centralized file storage requires that we have at least by default the index inside a segment.

To have a well working configuration property editor we need to add some more gui related information into the nutch configuration file. So we need the property type (string / integer) and a possible value range also as a default value and a validation regular expressions. These information can be added into the nutch-default.xml by adding some sub nodes to the existing property nodes.

...

## Pre Requirements:

- Be able to start all nutch tools like the fetch tool by an api call. This requires some minor changes in some tools that contain some simple logic processing already in the main method.
- Api based job starting from a container and also the ability to have multiple instances running in one jvm require the ability to pass a nutch conf instance within the call stack.

- working folder centralized data storage. For example storing index inside a segment folder, to be able to connect indexes and segement data physically.
- Segment status tracking is required to allow or prohibit functionality in the admin gui. This can simple realized by writing status identify files into the folder as partly already done with index.done.
- add gui related information like, valueType, validationPattern, defaultValue to configuration file.

## To discuss:

Starting processes in seperated JVMs as tasktracker does? Split configuration file into pieces of add category node.

## Timetable:

TODO

## people:

The Apache Nutch Development team.

Original developers working on this included Frank Hanze (jsp programming), Marko Bauhard (re-factoring nutchConf and tools api) & Stefan Groschupf (developing plugin Extension point and ground framework)

Please add yourself here.

If you wish to help but do not know how, please get in touch with the Nutch team here.

## Download:

The code base we are working on is Nutch 2.0, which you can checkout here. If you are unfamiliar with using SVN repositories and SVN, then please see here. Additionally, a now dated but fully functioning Nutch version of the examples shown in this page can be found (and checked out) here

## Old Resources

Here are some mirrors where you can download a version of nutch-0.8-dev bundled with the administration GUI, some of these mirrors no longer exist, and are there merely to provide you with a look and feel for the GUI.

- http://85.214.26.67/nutch-admingui/nutch-0.8-dev_guiBundle_05_02_06.tar.gz
- http://jerome.charron.free.fr/nutch/nutch-0.8-dev_guiBundle_05_02_06.tar.gz
- http://www.frutch.org/download/nutch-0.8-dev_guiBundle_05_02_06.tar.gz
- http://68.178.249.66/nutch-admin/nutch-0.8-dev_guiBundle_05_02_06.tar.gz
- http://www.bachansoft.com/mirror/nutch-0.8-dev_guiBundle_05_02_06.tar.gz