WhyNutchHasAPluginSystem

Originally written by StefanGroschupf - 05 Oct 2004, however the ethos and legacy behind the plugin system are still relevant.

This text explain the ideas behind the nutch plugin system.

There are 3 main arguments for a plugin system.

Extensibility

Plugins allow anyone to extend the functionality of Nutch simply by writing their own implementation of a given interface. For instance, the HTMLParser, used for parsing HTML documents, is an implementation of the Parser interface.

Flexibility

Since everybody can write a plugin, hopefully in future there will be a large set of plugins to choose from. Advances in the adaptability of the TIKA plugin to deal with many common file types has permitted the removal of a lot of clutter from the plugins which are distributed with Nutch. This allows Nutch administrators to assemble their own search engine based on her/his particular needs needs by installing the plugins he or she is interested in with little or no hastle. Administrators are also able to choose from different summarizing algorithms, add pdf file format or remove ftp protocol support.

Maintainability

Each developer is focused on his/her own context. The core developers are able to write code for the nutch core engine and provide a described interface - a plug. A plugin developer is able to focus on the functionality of their specific plugin without worrying about how the system as a whole works. They only need to know what kind of data the plug and the plugin exchange. Since both sides are encapsulated nobody needs to take care of the integration of the other side. This makes the code less complex, easier to maintain and less fragile.

<<< PluginCentral