

URLMap – a quick tutorial

<http://biit.cs.ut.ee/urlmap>

URLMap is a tool for mapping user queries to web-based tools and databases. In URLMap parlance, these objects are called *targets*. Targets are organized into categories and categories may be nested. Therefore, URLMap's main data structure is an arbitrary tree.

Administrative interface

This section of the application is only available to administrators (currently all BIIT members), and allows creating the hierarchy of categories and targets. Categories may be added, renamed, moved and deleted. The core of URLMap, though, are targets.

Handling targets

If not already there, open the administrative interface by clicking the link on the front page. The paged view of the root category appears. First of all, we need to navigate to a suitable spot in the category hierarchy and click “Add target” on the toolbar. When adding a target, some properties need to be specified. These are discussed below. As an example, we'll work through creating a simple Google search target.

- Title – The brief label for a target, this might be “Google search”.
- Description – An optional longer synopsis for the tool or database.
- Request method – The HTTP method used by the external database. URLMap supports both GET and POST, this should be GET for Google.
- Query modifier regex or code filter – An incoming query might need to be modified before actually using it. For example, some databases need symbols delimited by commas, but URLMap uses the convention of separating symbols with spaces. Therefore, let's use Perl regular expressions to change all sequences of whitespace in the query to commas. This is achieved with the regex

```
s/\s+/,/g
```

For very advanced manipulations, a full Perl code filter is available. The code filter receives three variables: \$QUERY, \$URL and \$FIELDS, which represent the query string, the URL template string and a hash reference of additional fields. These can then be manipulated in any conceivable fashion, but this is probably rarely if ever necessary and outside the scope of this tutorial. The code runs in a restricted sandbox though, so the underlying system is theoretically never compromised.

- URL – The web address template of the target. This string can contain special tokens delimited by hashes (#) which are substituted by elements of the query. For example, the main query field is always named “QUERY”. Constructing a target for searching Google, would therefore yield a URL string

```
http://google.com/search?q=#QUERY#
```

Every time this target is used, the token is replaced by the actual user query, therefore generating a link to Google's search results.

- Query tokenizing – Usually, a query string is passed to the mapping routine unchanged. This option allows to “chop up” the string into smaller units – for example, lists delimited by an empty line or single symbols. Assuming we have a query “awk sed grep” the Google target created earlier will usually yield a link looking like

```
http://google.com/search?q=awk+sed+grep
```

but using single symbol mapping, there will be 3 results:

```
http://google.com/search?q=awk
http://google.com/search?q=sed
http://google.com/search?q=grep
```

Don't select a tokenizer for now.

- Additional fields – A target can have extra fields in addition to the main query field. These can take the form of a text string (textbox), a boolean value (checkbox) or a predetermined list of allowed values (combobox). The specified token is then replaced with the actual value within the URL template at mapping. For example, we could augment the Google target with a field for specifying the language of the results page. This is controlled by the parameter “hl” which has a value of “en” for English and “et” for Estonian. A drop-down box would be suitable for this purpose. This field has three properties:
 - Replace token – A name specifying which part in the URL template to replace. We might

use "LANGUAGE".

- Label – The user-visible label describing this field. This might be "Interface language"
- Label=value pairs – Comma-separated labels and values for this drop-down box. This should contain "Estonian=et, English=en"

After these changes, the URL template can be modified to read

```
http://google.com/search?q=#QUERY#&hl=#LANGUAGE#
```

Other fields differ slightly, but should be easily understandable using this tutorial.

After clicking "Save", the new target is created.

Mapping

At this point, open the URLMap front page in a new window or tab. The tree should now contain the new target. After selecting the target, the additional field for selecting a language appears on the right. Pick one, and also enter a query consisting of a few words into the main text box, then click "Generate links". This generates the mapping results page. Click the generated link – a Google search page with the entered query, modified to use commas as delimiters and using the selected language should open up.

At the top there's a breadcrumbs navigation bar. Clicking "Front page with this query" takes us back to the main page with the query state restored. Go back to the target editor that was left open in a background tab, select the single symbol tokenizer, and save the target. Returning to the front page, click the little arrow next to the language selection field – this allows the user to select many values instead of just one. Select both "English" and "Estonian", and generate links again.

This time, there are many results even if we only selected one target. The reason is that now the query is interpreted as single symbols – every word in the query generates one link. But we have also selected two language parameters: therefore all combinations of words and languages have been generated – in effect, a matrix of results. Suppose the query was "awk sed grep", the results matrix could be visualised as

query: "awk" language: Estonian	query: "sed" language: Estonian	query: "grep" language: Estonian
------------------------------------	------------------------------------	-------------------------------------

query: "awk" language: English	query: "sed" language: English	query: "grep" language: English
-----------------------------------	-----------------------------------	------------------------------------

Also notice that the results table can be sorted by target, category or description clicking on the column header.

Other useful features

- **Search engine** – On the main page, in the tree viewer pane, is a search field. Try searching for some items – the results will be selected in the tree. Clicking "Options" unveils some features – the item types to search, regular expression support and generating links straight to the administrative interface.
- **Bookmarks** – Simple, session-based bookmarks are supported. Select some items in the tree and click "Add" in the bookmarks pane. Generate a few bookmarks, use and delete them. The bookmarks are stored in a cookie and not accessible to other users.
- **Moving and reordering items** – In the administrative interface, items in a category can be reordered using the red and green arrows next to the item. Clicking "Save" stores the new configuration. Categories can be moved – click "Move category" anywhere below the root level to select a new parent for the current category. Targets may be assigned to several categories: navigate to a target page and click "Assign categories" to select parents for the target.
- **Compact links** – On the mapping page, there's a button named "Generate compact link" on the toolbar. This yields a result page with a much prettier short URL. All parameters are saved on the server. But what makes this feature useful is that the saved parameters can be overloaded, therefore taking a compact link such as

`https://biit.cs.ut.ee/urlmap/urlmap.cgi?clink=a80f1c752d41cf47c16488a8adf017ae`

we can use a different query than saved on the server by appending it to the link:

`https://biit.cs.ut.ee/urlmap/urlmap.cgi?clink=a80f1c752d41cf47c16488a8adf017ae&p_query=my_query_goes_here`