EXT: Google Query (Data Provider)

Extension Key: googlequery

Language: en

Keywords: forAdmins, forintermediates

Copyright 2000-2010, Roberto Presedo, <typo3@cobweb.ch>

This document is published under the Open Content License

available from http://www.opencontent.org/opl.shtml

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

Introduction

What does it do?

This extension is a Data Provider for the Tesseract Project. This Data Provider retrieves information from the "Google Site Search" service (hereinafter "**GSS**") or a stand-alone "Google Search Appliance" solution (hereinafter "**GSA**"). It then returns the resulting data as a "recordset"-type Standardized Data Structure (see extension "tesseract" for a general explanation).

This extension also provides a TYPO3 plugin to display a search form. Combined with a "GSA", the search form can provide some added value features such as search suggestions.

What does it look like

Here is the typical input screen of a Google Query element. In the first tab you can set information about the Search engine.

Using a GSA search engine ...

or using the GSS service

In the second tab, you can list the data you want to retrieve from GSA

In the last tab you can configure the cache duration

Questions ?

If you have any questions about this extension, you may want to refer to the Tesseract Project web site (http://www.typo3-tesseract.com/) for support and tutorials. You may also ask questions in the TYPO3 English mailing list (typo3.english).

Keeping the developers happy

If you like this extension, do not hesitate to rate it. Go the Extension Repository, search for this extension, click on its title to go to the details view, then click on the "Ratings" tab and vote (you need to be logged in). Every new vote keeps the developers ticking. So just do it!

You may also take a step back and reflect about the beauty of sharing. Think about how much you are benefiting and how much you are is giving back to the community.

Configuring the Data Provider

The GoogleQuery data provider can be configured to send requests to a "Google Site Search" service (hereinafter "**GSS**") or a stand-alone "Google Search Appliance" solution (hereinafter "**GSA**")

In the first tab you must choose which search engine is going to be used.

Querying the "Google Site Search" service

About the "Google Site Search" service
Google provides two online "Custom Search Engine" services, but this extension can only interact with one of them. **"Google Custom Search" - (GCS)** With GCS you can create a customized search engine for your own website. This service is free, but unfortunately does not provide XML output. As the GoogleQuery data provider parses XML results provided by the Google tools (GSS or GSA), **the GCS is not compatible with GoogleQuery.** **"Google Site Search" - (GSS)** GSS is a hosted search solution which extends GCS. This solution isn't free (starting 100$USD per year) but provides XML output, which **makes the GSS compatible with GoogleQuery**. To create a new GSS visit http://www.google.com/sitesearch/ .

"General" tab

– Search Engine

Here choose "Google Site Search"

– Search engine unique ID

Insert here the Search engine unique ID provided by Google. You can find it in the control panel of your search engine, on http://www.google.com/cse/manage/all

This is a string like 999999999999999999999:srraaaaaaaa

"Advanced" tab

– Cache duration

If you want to use the Tesseract cache option, just set the cache duration (number of seconds). If this is set to 0, no cache will be used.

Querying a "Google Search Appliance" solution

About the "Google Search Appliance" solution
"Google Search Appliance" is a rack-mounted device providing document indexing functionality using a Google search-like interface for end-user retrieval. This device is supplied in two models : **"Google Mini" and "Google Search Appliance".** **Both devices are compatible with GoogleQuery**., although some features are available only with the "Google Search Appliance".

"General" tab

– Search Engine

Here choose "Google Search Appliance / Google Mini"

– Google Search Appliance result's url

Your Google Search Appliance host name or IP address, with the path to the search page (usually "/search").
NO SLASH AT THE END

Example : http://search.mycompany.com/search

– Frontend

The name of the frontend that the GSA must use.

A front end is a framework that manages most of the elements of a search process in a GSA, including the appearance of search and results pages, the data that is returned in search results or the arrangement of the search results .

As the appearance of search and results pages is not going to be generated by the GSA but by TYPO3, the setting of the ouput format in GSA is useless. However, other options can be set in Frontends such as KeyMatches and Related Queries settings, or removing special URLs from search results.

By default the GSA configuration comes with a Front End called default_frontend.

For more information about Front Ends in the GSA :
http://code.google.com/apis/searchappliance/documentation/68/help_gsa/serve_frontends.html
or
http://code.google.com/apis/searchappliance/documentation/46/help_mini/serve_frontends.html

– Collection

The name of the collection that will be used by the GSA for the request.

In the GSA you can create collections of documents that are subsets of the complete index. Each collection is defined by a group of URL patterns that encompasses the URLs of the documents in the collection.

– Type of information returned

Choose the kind of information that is going to be returned to the Data Consumer. Five options are available here.

- **Search results**
  List of all documents found by the GSA

- **Spelling suggestions**
  List of all spelling suggestions provided by the GSA

- **Related Queries**
  List of all related queries provided by the GSA

- **Keymatches**
  List of every keymatches provided by the GSA

- **Configured by Typoscript**
  You can also configure this information using Typoscript (see chapter "Configuration via Typoscript")

"Meta tags" tab – Only for GSA

About Meta Tags
HTML pages can contain meta tags in their source code that provides additional information on the content of the page. This information is recorded by the GSA, and available in the search results. It is possible to tell the GSA to only find documents that contain specific meta tags.

– Required metatags

Only documents containing those tags will be returned.

You can set as many required meta tags as you want, but enter one tag per line.

This is optional

– Selected metatags

This is the list of every tags that could be mapped in a Data Consumer.

You can set as many required meta tags as you want, but enter one tag per line.

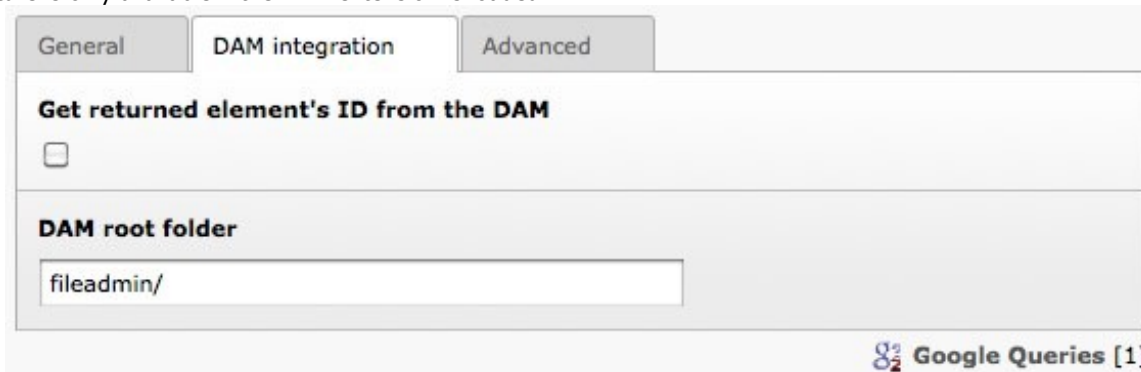This list is optional.

"Advanced" tab

– Cache duration

If you want to use the Tesseract cache option, just set the cache duration (number of seconds). If this is set to 0, no cache will be used.

Using GoogleQuery as a secondary data provider

The following features are only available if GoogleQuery is used as a secondary provider. To understand the difference between a primary and a secondary data provider, please read the Tesseract documentation.

"DAM integration" tab

This feature is only available if the DAM extension is loaded.

- Enable DAM integration

You can tell GoogleQuery to return the tx_dam.uid of the document found instead of the document url.

As a secondary data provider, **the list of DAM uids will be passed to the primary data provider**, which can be a simple dataquery that is going to check access rights to the documents found.

- DAM root folder

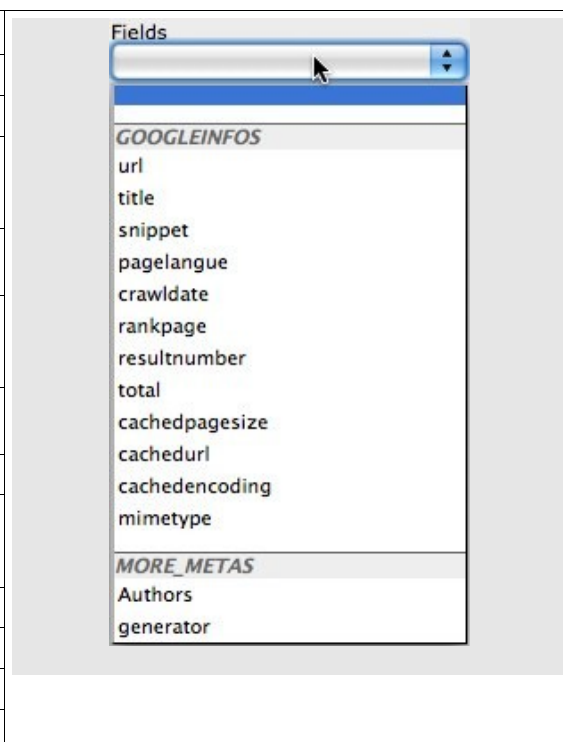Path to the DAM folder

Mapping results in a data consumer

At least 4 different kind of information is passed to Data Consumers elements, like TemplateDisplay:

- ○ Search results
- ○ Spelling suggestions
- ○ Related Queries
- ○ Keymatches

"Search results" for GSS and a GSA

Here is the list of fields returned to the Data Consumer by GoogleQuery, if you use GSS or if you selected "search results" as "Type of information returned" in the GSA configuration.

Field name	Description
url	The URL of the search result.
title	The title of the search result.
snippet	The snippet for the search result. Note: Query terms appear in bold in the results. Line breaks are included for proper text wrapping.
pagelang	Indicates the language of the search result. The LANG element contains a two-letter language code.
crawldate	Shows the date when the page was crawled. It is shown only for pages that have been crawled within the past two days.
rankpage	Provides a general rating of the relevance of the search result.
resultnumber	The index number of this search result
total	The estimated total number of results for the search. The estimate of the total number of results for a search can be too high or too low.
cachedpagesize	The size of the cached version of the search result.
cachedurl	The url of the cached version of the search result.
cachedencoding	The encoding of the cached version of the search result.
mimetype	The mime type of the search result.

Fields

GOOGLEINFOS
url
title
snippet
pagelangue
crawldate
rankpage
resultnumber
total
cachedpagesize
cachedurl
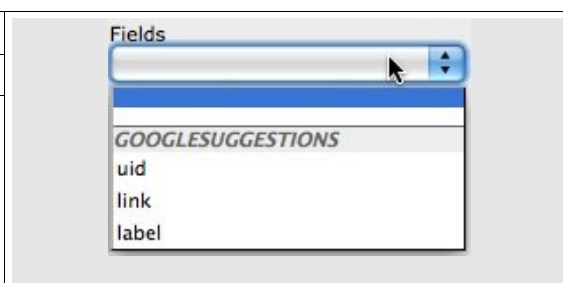cachedencoding
mimetype

MORE_METAS
Authors
generator

"Spelling suggestions" in a GSA

This is only available when using a GSA

Here is the list of fields returned to the Data Consumer by GoogleQuery if you selected "Spelling suggestions" as "Type of information returned" in the GSA configuration.
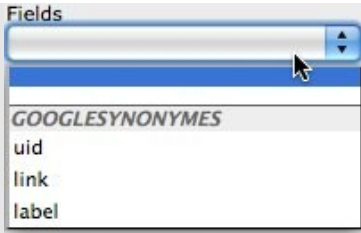
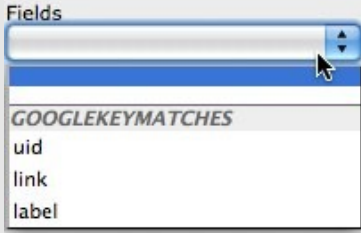| Field name	Description
label	Label of the spelling suggestion provided by the GSA.
link	Value for the GET q parameter for the current spelling suggestion

Fields

GOOGLESUGGESTIONS
uid
link
label

"Related Queries" in a GSA

This is only available when using a GSA

Here is the list of fields returned to the Data Consumer by GoogleQuery if you selected "Related Queries" as "Type of information returned" in the GSA configuration.

Field name	Description
label	Label of the related query provided by the GSA.
link	Value for the GET q parameter for the current related query.

"Keymatches" in a GSA

This is only available when using a GSA

Here is the list of fields returned to the Data Consumer by GoogleQuery if you selected "Keymatches" as "Type of information returned" in the GSA configuration.

Field name	Description
label	Name of the keymatch.
link	Full url of the current keymatch.

Configuring the plugin (search form)

Selected Plugin

g? GoogleQuery Search Form ▼

Plugin Options

Search Form

Page displaying GoogleQuery results

GSA Search Results ☒ ▢

📄 Page

Template File

✏️

Search Engine

Google Search Appliance / Mini ↕

Google Search Appliance host

Frontend

Collection

Enable "Auto-Suggestion"

☐

CSS file for the "Auto-Suggestion" option

Enable "Advanced Search Reporting"

☐

Why a separated search form plugin

Tesseract allows you to build any kind of page, including search forms. So why using a dedicated plugin to build a search form ?

Google Search Engines have some features that need to interact with the source code of the page rendering results. In the Tesseract project, the generated source code is not managed by a Data Provider, but by a Data Consumer. Thus, the extension GoogleQuery has no way to interact with the source code generated.

To simplify the task of creating a search engine, a plugin is provided with this extension to easily create a simple search form which can also display, for example, search suggestions provided by a GSA.

How to configure the plugin

Configuration for GSS and GSA

Page displaying GoogleQuery results

Select the page that will display the search results. If empty, the current page will be used as target.

Template File

HTML template file. If empty, the default template file will be loaded (EXT:/GoogleQuery/pi1/res/template.html).

Search Engine

Select the type of Google Search engine that will return the search results, GSS or GSA

Options for GSA

Google Search Appliance host

The root url of the Google Search Appliance. Example http://mygsa.mysite.com or http://123.123.123.123

Frontend

This is the name of the frontend used to display results. This is use to build the "Auto-Suggestion" or the "Advanced Search Reporting" (if enabled - see below).

THIS IS UNRELATED TO THE GoogleQuery DATAPROVIDER'S FRONTEND.

Collection

This is the name of the collection used to display results. This is use to build the "Auto-Suggestion" or the "Advanced Search Reporting" (if enabled - see below).

THIS IS UNRELATED TO THE GoogleQuery DATAPROVIDER'S COLLECTION.

Enable "Auto-Suggestion"

The "Auto-Suggestion" provides search suggestions to the users while typing.

The "Search-as-you-type" feature is only available on GSA Software version 6.0 and more.

CSS file for the "Auto-Suggestion" option

Name of the CSS file that stylizes the suggestions list provided by the "Search-as-you-type" feature. If empty, the default CSS file will be loaded (EXT:GoogleQuery/pi1/res/css/autosuggest.css)

Enable "Advanced Search Reporting"

The "Advanced Search Reporting" allows the GSA to log which links are clicked by the users on the result page. GSA will use this information to update the page's rank.

The "Advanced Search Reporting" feature is only available on GSA Software version 5.2 and more.

How to configure the template file

Default HTML template for the search form is located in the extension's `pi1/res/` directory.

The following subparts are defined on the top level:

Subpart	Description
GoogleQuery_SEARCHFORM	This subpart defines the markup for the simple search form (without "auto-suggestion" or "Advanced Search Reporting")
GoogleQuery_SEARCHFORM_SS	This subpart defines the markup for the extended search form (with "auto-suggestion" and "Advanced Search Reporting")

The following table describes subparts inside the GoogleQuery_SEARCHFORM top level subpart

Subpart	Description
GQ_FORM_TARGET	URL of the search page results
GQ_FORM_ID	Id of the form element
GQ_Q	Value of the $_GET['q'] variable
GQ_LABEL_SEARCH	Localized label for the "Search" button

The following table describes subparts inside the GoogleQuery_SEARCHFORM_SS top level subpart

Subpart	Description
GQ_FORM_TARGET	URL of the search page results
GQ_FORM_ID	Id of the form element
GQ_Q	Value of the $_GET['q'] variable
GQ_LABEL_SEARCH	Localized label for the "Search" button
GQ_GSA_JS	All JS code related to the "auto-suggestion" and "Advanced Search Reporting" features.
GQ_CLICKLOG	JS code related to the "Advanced Search Reporting" option. Is used only if "Advanced Search Reporting" is enabled

Configuration using Typoscript

Include static extension templates

This step is optional, as you can fully use the GoogleQuery extension without configuring Typoscript. But if you want to set default parameters for all GoogleQuery's instances, Typoscript is the solution.

The best way to use Typoscript with GoogleQuery is to include the static templates provided by this extension. To do so, open your main TS-template (in list view), click on the "Includes" tab and include the "GoogleQuery" template. If you dont't know how to do this -> see section "Templates" in the "Getting Started" document: http://typo3.org/documentation/document-library/tutorials/doc_tut_quickstart/0.1.0/view/1/9/
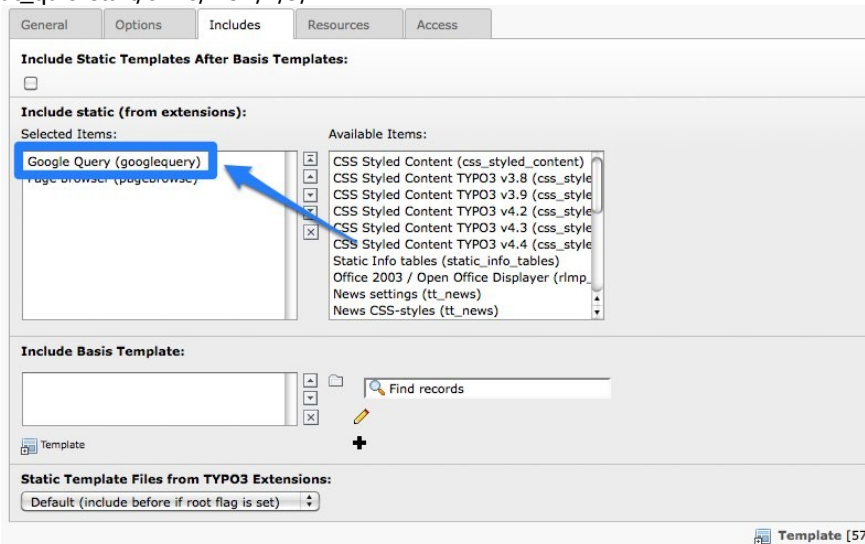
Constants

General Configuration

In the "Template" module, use the Constant Editor and choose the "GoogleQuery" category. Here you can set the following options.

Field name	Description
plugin.tx_GoogleQuery_pi1.searchEngineType	Search Engine Type gsa : Google Search Appliance/Mini gss : Google Site Search
plugin.tx_GoogleQuery_pi1.templateFile	Search form template - Name of the html template file for the search form
plugin.tx_GoogleQuery_pi1.targetId	Page results - Page id where GoogleQuery results are displayed
plugin.tx_GoogleQuery_pi1.cache_duration	Cache duration - Number of seconds

GSA Configuration

In the "Template" module, use the Constant Editor and choose the "GoogleQuery_GSA" category. Here you can set the following options.

Field name	Description
plugin.tx_GoogleQuery_pi1.gsa_host	GSA Host - Example : http://mygsa.mysite.com DO NOT ADD SLASH AT THE END
plugin.tx_GoogleQuery_pi1.collection	Name of the collection
plugin.tx_GoogleQuery_pi1.frontend	Name of the frontend
plugin.tx_GoogleQuery_pi1.autoss	"Auto-Suggestion" - Provides search suggestions when typing in the search form (See the "About some special GSA features" section)
plugin.tx_GoogleQuery_pi1.autosscss	"Auto-Suggestion" CSS file - CSS file for the "autoss" option
plugin.tx_GoogleQuery_pi1.clicklog	"Advanced Search Reporting" - Allows the GSA to log which links are clicked by the users on the result page (See the "About some special GSA features" section)

Field name	Description
plugin.tx_GoogleQuery_pi1.metatags_required	Metatags required - Only documents containing this meta tags will be returned - Comma separated values
plugin.tx_GoogleQuery_pi1.maintable	Type of information to return - Here you can specifiy the kind of information that is going to be returned by the GSA googleInfos : search results googleSuggestions : spelling suggestions googleSynonymes : related queries googleKeymatches : keymatches This value is going to be used by the Data Consumer to make a loop on it. See tesseract documentation about maintables.

GSS Configuration

In the "Template" module, use the Constant Editor and choose the "GoogleQuery_GSS" category. Here you can set the following option.

Field name	Description
plugin.tx_GoogleQuery_pi1.gss_id	GSS Search engine unique ID Find it on your search engine Control panel http://www.google.com/cse/manage/all/

Setup

Here is the setup.txt file loaded

```
plugin.tx_GoogleQuery_pi1{
    templateFile = {$plugin.tx_GoogleQuery_pi1.templateFile}
    targetId = {$plugin.tx_GoogleQuery_pi1.targetId}
    autoss = {$plugin.tx_GoogleQuery_pi1.autoss}
    autosscss = {$plugin.tx_GoogleQuery_pi1.autosscss}
    gsa_host = {$plugin.tx_GoogleQuery_pi1.gsa_host}
    frontend = {$plugin.tx_GoogleQuery_pi1.frontend}
    collection = {$plugin.tx_GoogleQuery_pi1.collection}
    searchEngineType = {$plugin.tx_GoogleQuery_pi1.searchEngineType}
    gss_id = {$plugin.tx_GoogleQuery_pi1.gss_id}
    clicklog = {$plugin.tx_GoogleQuery_pi1.clicklog}
}
config.tx_GoogleQuery_queries {
    server_address = {$plugin.tx_GoogleQuery_pi1.gsa_host}/search
    client_frontend = {$plugin.tx_GoogleQuery_pi1.frontend}
    collection = {$plugin.tx_GoogleQuery_pi1.collection}
    maintable = {$plugin.tx_GoogleQuery_pi1.maintable}
    cache_duration = {$plugin.tx_GoogleQuery_pi1.cache_duration}
    searchEngineType = {$plugin.tx_GoogleQuery_pi1.searchEngineType}
    gss_id = {$plugin.tx_GoogleQuery_pi1.gss_id}
}
config.tx_GoogleQuery_queries2 < config.tx_GoogleQuery_queries
```

What about metatags ?

Metatags are the only information that **cannot be defined using TypoScript**. The reason is that unlike other information, metatags is used both on frontend (to make the list of required metatags) and backend (to provide the list of metatags to the data consumer, in order to map those metatags). As long as TypoScript is only available on Frontend, we cannot set it this way.

About some special GSA features

"Auto-suggestion"

The "Search-as-you-type" feature is only available on GSA Software version 6.0 and more.

What is this ?

The query suggestion service provides suggestions that complete a user's search query. As a user enters a query in the search box, a drop-down menu appears with suggestions to complete the query. The search appliance uses the most popular search queries of its users to determine the top suggestions that list for a query. In addition, if activated, the search appliance adds user-added results to the list of suggestions.

How does it works ?

The GoogleQuery search form plugin loads a JavaScript file (/pi1/res/autosuggest.js).

When a user starts a query, the JavaScript in the client makes calls to the query suggestion URI and fetches the results, responding with JSON output. The AJAX response handler in the JavaScript client populates the list of suggestions.

Hot to configure this on your TYPO3 search pages ?

You just have to enable "Auto-Suggestion option in the plugin configuration (in the plugin itself or using Typoscript), and that's all. Using the default search form template, everything is ready to provide the list of suggestions.

You can also change the design of the drop-down menu by changing the CSS file for the "Auto-Suggestion" option in the plugin configuration (in the plugin itself or using Typoscript).

Where to get more information

If you need more detailed information, read
http://code.google.com/apis/searchappliance/documentation/60/xml_reference.html#QuerySuggestionsSuggestURLParameter

"Advanced Search Reporting"

The "Advanced Search Reporting" feature is only available on GSA Software version 5.2 and more.

What is this ?

Advanced search reporting enables administrators to see what types of links a user chooses on a search results page, and more generally to track all actions that a user performs such as clicking navigational links. This information enables administrators to improve access and latency of search results, and to understand users click-behavior.

How does it works ?

When you enable this feature, the GoogleQuery extension modifies search form by inserting Javascript that tracks all links that a user clicks. When a user clicks on a link in the search results, Javascript executes in the browser and requests an URL on the search appliance (which starts with "/click") that contains information about the link. The search appliance logs the arguments given in the URL on the search appliance and then returns a response to the browser. The browser then retrieves the URL on which the user clicked.

How to configure this on your TYPO3 search pages ?

As the search page may contain a great number of links not related to the search results (navigation menu's links, footer's links, ...), Javascript will track only links configured to be tracked.

The following link attributes should be added to the tracked link. All of them are required

Attribut	Description
`rel`	This must be "logclick" Only links with rel="logclick" will be tracked
`ctype`	Click type. A value that identifies the type of link that a user clicks. For the value, use underscores or a dot without spaces and use alpha-numeric characters. For a complete list of click types, see http://code.google.com/apis/searchappliance/documentation/52/admin_searchexp/ce_improving_search.html#clicktypes
`pos`	Rank of the result on which the user clicks. First link = 0, Second link = 1, ...

Example of link :

```
<a ctype="c" pos="2" rel="logclick" href="http://my.link.com">The link</a>
```

By clicking on that link, the GSA will log that the user

- clicked in the search result list (ctype=c),

- on the third link (pos=2),

- which was pointing to http://my.link.com (href="http://my.link.com"),

- while searching for a query text (automatically detected by the Javascript)

The GSA call that Javascript makes is : http://my.gsa.com/click?ct=c&r=3&url=http%3A//my.link.com&q=my+query

Where to get more information

If you need a more detailed information, read
http://code.google.com/apis/searchappliance/documentation/52/asr_reference.html#Request_Parameters

Quick start : a Google search engine in 4 steps

GoogleQuery provides a t3d file which contains examples of search forms and search results both for GSS and GSA solutions.

Requirements

The t3d file will install different elements (pages, template files, datafilters element, displaycontroller elements, …). Before installing all these elements, some extensions must be installed. Here is the list of Tesseract extensions required if you want to successfully install these examples.
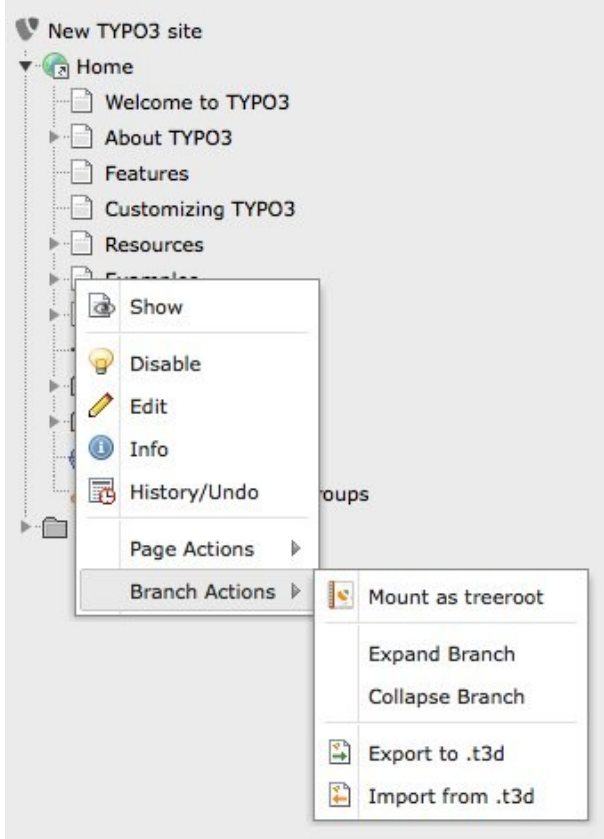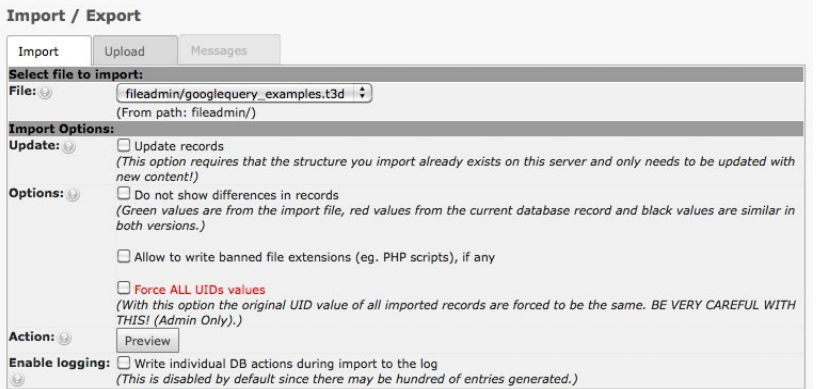
Please install these extensions in the following order :
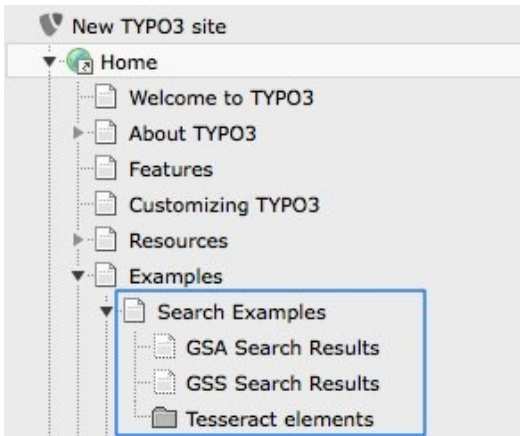
1. expressions
2. tesseract
3. overlays
4. displaycontroller
5. datafilter
6. templatedisplay
7. pagebrowse
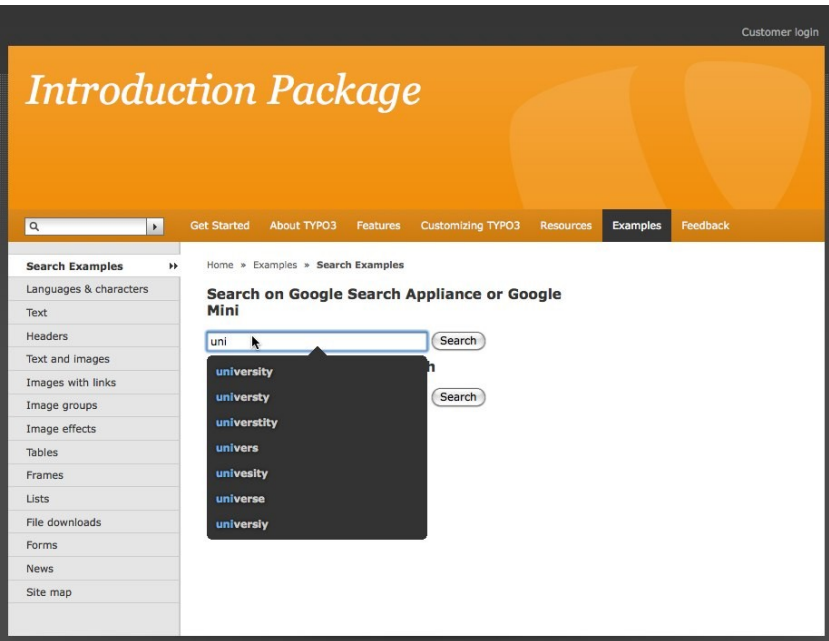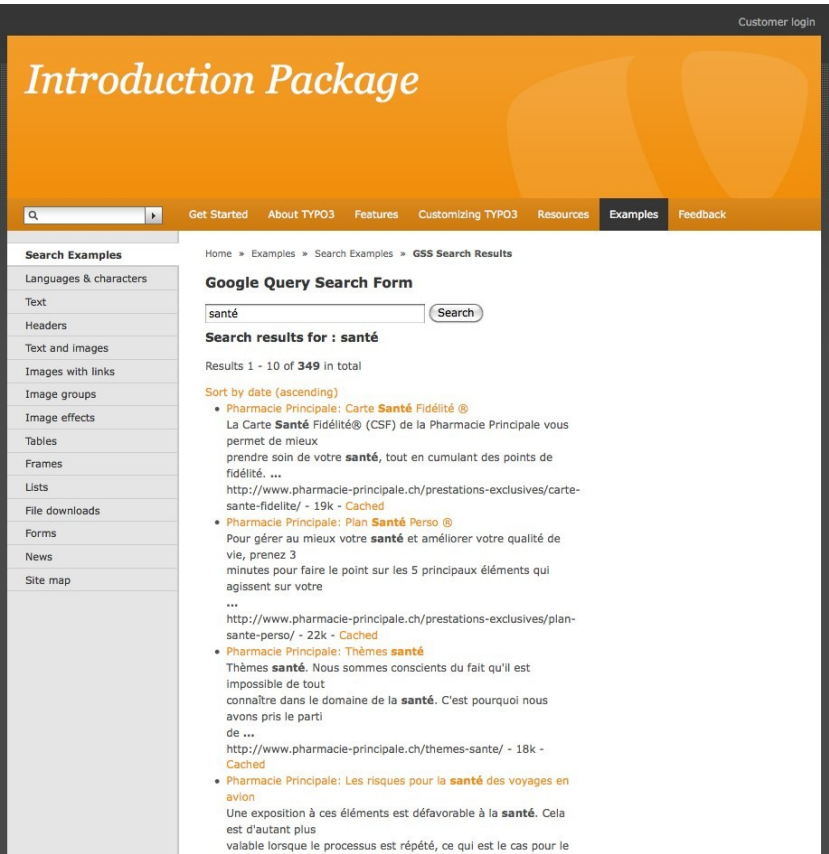8. GoogleQuery (Advice : enable the debug feature in the extension configuration)

Note : During the configuration of Tesseract extension, the extension provides a quick way to check if the main extensions that make up Tesseract are installed. Tesseract is of a modular nature. This means that you may perfectly run without some of the suggested extensions. You can install all extensions suggested by Tesseract, but **to run the examples provided by GoogleQuery, only the extension list above is required.**

Install the examples pages

Note : The screenshots below were taken on a brand new instance of the Introduction Package provided by TYPO3 (version 4.5.2). Nothing else has been added to the package but the 8 extensions listed on the previous page.

Step	Screenshot
Step 1 Copy GoogleQuery_examples.t3d in the fileadmin folder. This file is located in /res/GoogleQuery_examples.t3d. **Import the t3d file**. Open the Typo3 page tree. You will need to have at least one page there. If not, you must create it now. Then left-click onto the start page, after which you want to restore the file into it. Select 'Branch Actions' - 'Import from .t3d'. Select the GoogleQuery_examples.t3d file you previously copied in the fileadmin folder and import it.	

Step	Screenshot
Step 2 Check if the structure is correctly created **Four new pages should have been created** : *Search Example* contains two search forms : one to query a GSA, the other to query a GSS service *GSA Search Results* displays a displaycontroller element configured to output GSA results *GSS Search Results* displays a displaycontroller element configured to output GSS results *Tesseract elements* sysfolder containing all Tesseract elements such as datafilters and templatedisplays	New TYPO3 site ▼ Home Welcome to TYPO3 ▶ About TYPO3 Features Customizing TYPO3 ▶ Resources ▼ Examples ▼ Search Examples GSA Search Results GSS Search Results Tesseract elements
Step 3 On "Search Examples" page, use Template Module to set constants. **To set the GSA search engine,** choose category "GoogleQuery_GSA" and set the GSA Host [plugin.tx_GoogleQuery_pi1.gsa_host] **To set the GSS search engine,** choose category "GoogleQuery_GSS" and set the GSS Search engine unique ID [plugin.tx_GoogleQuery_pi1.gss_id]	**Edit constants for template:** **Tesseract Configuration** Category: GOOGLEQUERY_GSA (8) **GSA Host** [plugin.tx_googlequery_pi1.gsa_host] Example http://mygsa.mysite.com ✏ [Empty] **Edit constants for template:** **Tesseract Configuration** Category: GOOGLEQUERY_GSS (1) **GSS Search engine unique ID** [plugin.tx_googlequery_pi1.gss_id] Find it on your search engine Control panel (http://www.google.com/cse/manage/all/) ✏ [Empty]

Step	Screenshot
Step 4 In frontend, open the "Search Examples" page and **start searching something.** Enjoy	

Known problems

– Google search engines have some limitations. One is a limit of 100 maximum results returned. This means that if you use the Pagebrowse option in Tempate Display, the pagination may be wrong when displaying the results around the 100th position.

If you have any other issue, please refer to the Tesseract Project web site (http://www.typo3-tesseract.com/).

You may also post your problems to the TYPO3 English mailing list (typo3.english), so that others may benefit from the answers too.

For bugs or feature requests, please open an entry in the extension's bug tracker on Forge (http://forge.typo3.org/projects/extension-googlequery/issues).

To-Do list

The roadmap for the evolution of this extension can be found on Forge: http://forge.typo3.org/projects/roadmap/extension-googlequery

For feature requests, please open a report on Forge issue tracker: http://forge.typo3.org/projects/extension-googlequery/issues

ChangeLog

Version	Changes:
2.0.0	Second version including ○ **Typoscript** support ○ **"Google Site Search"** Support ○ **Spelling suggestions** using "Google Search Appliance" devices ○ **Related Queries** using "Google Search Appliance" devices ○ **Keymatches** using "Google Search Appliance" devices ○ **"Auto-suggestions"** feature using "Google Search Appliance" devices ○ "**Advanced Search Reporting"** feature using "Google Search Appliance" devices ○ **DAM integration** ○ **Customizable examples** in a t3d file
1.0.0	Initial public release