

Data Pump Search Interface

The Data Pump's search interface is located at the following URL:

`http://<search name>/archive/search`

It should be accessed through the GET method. Three variables in the query string are expected:

Name	Description
lang	Two-letter ISO language code. Either 'fi' or 'sv'.
terms	Terms that must appear in a data channel's contents. A channel is considered a match if all terms are found. Terms should be separated by commas.
expressions	Like terms, except that a list of comma-separated regular expressions is expected.
type	If specified, limit the search results by type. Either 'meeting' or 'history'.

The variable lang must be set and either terms or expressions must be present.

Search results are returned in a JSON object, which has a single element: channels. It's an array containing objects as described below:

Name	Description						
titles	An array containing the titles of a search result. A result matching a sub-section will have multiple titles. For example, an agenda item will have the title of the meeting followed by the title of the agenda.						
hash	URL hash that directs the browser to the sub-section in question. The prefix A means an agenda. The prefix T indicates a video time.						
channelName	Name of the channel. Could have one of the follow patterns: <table border="1"><tbody><tr><td><code>kvsto-<year>-<number>@<year>-<month>-<day></code></td><td>A channel containing the proceeding of a meeting. meeting.js should be used to display the contents.</td></tr><tr><td><code>hel-<year>-<number></code></td><td>A channel containing the history of an issue. history.js should be used to display the contents.</td></tr><tr><td><code>jasen-<last name>-<first name></code></td><td>A channel containing speeches made by a council member. member.js should be used to display the contents. If a name contains accented Latin letters, the diacritics are stripped.</td></tr></tbody></table>	<code>kvsto-<year>-<number>@<year>-<month>-<day></code>	A channel containing the proceeding of a meeting. meeting.js should be used to display the contents.	<code>hel-<year>-<number></code>	A channel containing the history of an issue. history.js should be used to display the contents.	<code>jasen-<last name>-<first name></code>	A channel containing speeches made by a council member. member.js should be used to display the contents. If a name contains accented Latin letters, the diacritics are stripped.
<code>kvsto-<year>-<number>@<year>-<month>-<day></code>	A channel containing the proceeding of a meeting. meeting.js should be used to display the contents.						
<code>hel-<year>-<number></code>	A channel containing the history of an issue. history.js should be used to display the contents.						
<code>jasen-<last name>-<first name></code>	A channel containing speeches made by a council member. member.js should be used to display the contents. If a name contains accented Latin letters, the diacritics are stripped.						

The results will be in reverse chronological order.

Example:

```

{
"channels": [
{
"titles": [
"Päätöshistoria HEL-2013-015159",
"Kaupunginhallitus 2014-04-22"
],
"channelName": "hel-2013-015159",
"hash": "D2014-04-22"
},
{
"titles": [
"Kaupunginvaltuuston kokous 10/2014.6.4",
"Lainan myöntäminen Helsinki Stadion Management Oy:lle"
],
"channelName": "kvsto-2014-10@2014-06-04",
"hash": "A8"
},
{
"titles": [
"Kaupunginvaltuuston kokous 9/2013.5.15",
"Länsisataman tontin 20803/1 asemakaavan muuttaminen (nro12101; Jätkäsaaren
tornihotelli)",
"Andersson Hennariikka (Kokoomus) 0:24"
],
"channelName": "kvsto-2013-9@2013-05-15",
"hash": "A6T10846"
}
]
}

```

Retrieving the actual data

The data of a given channel can be found at an URL with the following pattern:

`http://<server name>/archive/<channel name>/<language code>`