

Data-pump front end

Proceedings from a city council meeting is presented to web visitors with the help of JavaScript and Cascading Style Sheets. The scripts and CSS files are served by the same Nginx server that provides WebSocket broadcast. Any web page, regardless of domain name, can link in the files and display the contents.

The main script is hosted at the following URL:

`http://<server name>/js/meeting.js`

This script will in turn link in other necessary scripts (with the exception of jQuery) and CSS files. A single SCRIPT tag will insert contents about a meeting into a web page. Parameters, specifying the meeting and the language, are stored in a JSON object as the tag's text:

```
<script src="http://<server name>/js/meeting.js">
{
  "channelName": "kvsto-2013-10@2013-05-29",
  "languageCode": "fi",
  "containerSelector": "#container",
  "live": true
}
</script>
```

channelName	The data source identifier. The channel name for a city council meeting is formatted as follows: <code>kvsto-<year>-<meeting number>@<year>-<month>-<day></code> The numbers for month and day are zero-padded.
languageCode	Two letter language code ("fi" or "sv")
containerSelector	jQuery selector used to identify the HTML element where the contents is placed. The element will be emptied first.
live	Whether the feed is live (i.e. the meeting is ongoing).

Live data feed is transferred over WebSocket. The data broadcast is handled by the data-pump's Broadcaster component. For past meetings, data is transferred over regular HTTP. All items are packed into an array, then encoded and sent as JSON. This part of the data-pump's operation is handled by its Archivist component.

The URL to the JSON data is as follows:

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

Note: There should be no trailing slash in the URL.

You shouldn't need to concern yourself about how the URL is formed. The script file data-pump.js (pulled in by meeting.js) takes care of its formation.

Generation of Supplementary Contents

Besides acting as a data source for past meetings, the Archivist also serves certain documents that the meeting agenda points to. These include: text concerning an upcoming decision, the decision history of an issue, and voting maps.

The decision history page is generated in a manner that's exactly analogous to how the meeting page is generated: the data arrives as JSON which is then presented with the help of JavaScript code. The registry ID of the issue server as the channel name in this case, with spaces replaced by dashes and letters converted to lowercase. For example, the data of issue "HEL 2012-000918" is located at `http://<server name>/hel-2012-000918/fi`.

The issue history make use of the the following JavaScript:

`http://<server name>/js/history.js`

The following HTML snippet is used to pull in the issue history:

```
<script src="http://<server name>/js/history.js">
{
  "channelName": "hel-2012-000918",
  "languageCode": "fi",
  "containerSelector": "#container",
  "live": false
}
</script>
```

The following snippet yields a single event within the issue history:

```
<script src="http://<server name>/js/history.js">
{
  "channelName": "hel-2012-000918",
  "languageCode": "fi",
  "containerSelector": "#container",
  "live": false,
  "policymaker": "Kaupunginvaltuusto",
  "date": "2013-05-29"
}
</script>
```

policymaker	The policymaker that made or is making a decision concerning the issue. Will always be "Kaupunginvaltuusto" for our purpose.
date	The date the decision was or will be made.

A standalone copy of a voting map can be obtain using the following snippet (note how meeting.js is used here):

```
<script src="http://<server name>/js/meeting.js">
{
  "channelName": "kvsto-2013-10@2013-05-29",
  "languageCode": "fi",
  "containerSelector": "#container",
  "live": false,
  "issueNumber": "4",
  "votingNumber": "1",
  "grayscale": false
}
</script>
```

issueNumber	A number identifying the issue (4 means the fourth issue discussed in the meeting).
voteNumber	A number identifying the voting round (1 means the first round of voting that took place during the meeting).

Links to HTML Files

The Archivist will generate the snippets described above (plus <html>, <body>, etc.) when requests are made to URLs of the following form:

http://<server name>/archive/<channel name>/<language code>/html/<filename>.html

The filename determines which type of snippet gets generated. Parameters are also extracted from the filename. The following names are possible:

historia -<issue number>	The issue history. The issue number is only there so different issues handled in a meeting have different filenames. It's ignored since the issue is already identified by the channel name.
history -<issue number>	Same as above.
asia -<date>-<issue number>	Text about a decision concerning the issue to be made during the city council meeting. Date is the meeting date and has the format YYYY-MM-DD. The issue number is ignored.
fraga -<date>-<issue number>	Same as above.
issue -<date>-<issue number>	Same as above.
aanestys -<date>-<issue number>-<vote number>	Voting map in color. Date has the format YYYY-MM-DD. It's actually ignored since the meeting date is already present in the channel name.
rosthning -<date>-<issue number>-<vote number>	Same as above.
voting -<date>-<issue number>-<vote number>	Same as above.

aanestys -<date>-<issue number>-<vote number>-mv	Voting map in grayscale.
rostning -<date>-<issue number>-<vote number>-mv	Same as above.
voting -<date>-<issue number>-<vote number>-mv	Same as above.
kokous -<date>	The meeting proceedings.
samling -<date>	Same as above.
meeting -<date>	Same as above.

Links to PDF Files

The Archivist will also generate PDF versions of the files described above. Their URLs have the following form:

http://<server name>/archive/<channel name>/<language code>/pdf/<filename>.pdf

Replacing "pdf" with "html" yield the URL of the HTML version.

HTML Snapshots

Search engines typically have difficulty indexing pages containing contents generated by client-side script. To alleviate this problem, you can ask the Archivist to generate a HTML snapshot containing already rendered contents. Simple append /snapshot to the URL of a HTML page:

http://<server name>/archive/<channel name>/<language code>/html/<filename>.html/snapshot

To point the search engine back to the actual page, place its URL in the query variable **canonical**. A <link rel="canonical" href=" ... "> tag will be added to the resulting page.