Content source (aka CMS data delivery)

The content delivery service can serve any view for external systems (for example: your website).

The content interface serves in different formats (json,html,xml), and provides a caching mechanism too.

- Setting up a content source
- Consuming a Content Source

Setting up a content source

THIS ARTICLE IS UNDER DEVELOPMENT

Many parameters are straight forward

- Related solution
- Max cache age
- Is active

Unique NAME

This is the calling name of the interface, that must be provided accessing the content service.

Example for calling the view **myview**

```
.../cmsinterface?q=myview& ...
```

Query command

Relevant commands include

- General records
  - list: List of records
  - show: Single record
- HTML formatted views
  - heat: Heatmaps
  - gant: Gant charts
  - xtab: Pivot tables
  - calm: Calendar

Query parameters

Parameters are provided without URL encoding

```
QUERY_FIELD_1=TITEL&QUERY_OPERATOR_1=8&QUERY_VALUE_1=Hello&QUERY_SHOWFIELD=TITEL StatusID DEADLINE
```

Procedure for easy parameter setup

1. Build views in frontend.
2. Activate the link and copy the URL
3. ´Remove unneeded parameters
   - command
   - SagID
   - QUERY_NEW

Possible parameters

QUERY_FIELD_n	Part of the search/filter. The field to filter by.
QUERY_OPERATOR_n	Part of the search/filter. The way to filter. Full list Dashboard widget configuration
QUERY_VALUE_n	Part of the search/filter. The value to filter by.
QUERY_SHOWFIELD	The fields to show in the list, separated by space.
QUERY_PageSize	Number of records to show pr page.
QUERY_PageOffset	The page number to show.
QUERY_SortOrder	The field to sort ascending by. Use either this or SortOrderDesc.
QUERY_SortOrderDesc	The field to sort descending by. Use either this or SortOrder.
QUERY_Grouping	The field to group records by.
QUERY_GroupingDesc	

Dynamic parameter

This specifies a parameter that will be set by the value provided in the **v** parameter.

Example: Using this interface

```
Unique NAME: mylist
Query command: list
Dynamic parameter: QUERY_PageOffset
```

Making a call to

```
.../cmsinterface?q=mylist&v=2
```

Will fetch a list of data and display page 2, because **2** is injected into **QUERY_PageOffset**.

Parameters and values are injected directly into the http request.

Note: Using the "show" command the variable will ALLWAYS get mapped to "DataID"

Using a content source

Calling the **cmsinterface** servlet will provide you with an overview on how to use it.

Example:

```
https://www.acme.com/TempusServa/cmsinterface
```

The interface requires just the name of the source, but additional parameters can be provided.

Parameters

- **q**: Name of the interface (required)
  - This name must match "Unique NAME"
- **f**: Format of the result (optional)
  - Valid values: html json xml
- **v**: Parameter for dynamic values (optional)
  - The value will be substituted to the parameter in "Dynamic parameter"

Example:

```
https://www.acme.com/TempusServa/cmsinterface?q=mysource&f=json
```

Troubleshooting

THIS ARTICLE IS UNDER DEVELOPMENT

Configuration caching

For performance reasons information about the content sources is kept cached.

Adding or changing sources will propegate automatically : You will need to clear the system cache to pick the changes.

Parameter encoding

Note that URL's copied from a browser address bar are often encoded in HTML format.

The content will not decode values and request strings, so all encoded strings will fail.

Consuming a Content Source

CMS content provider

The CMS connector will allow CMS systems to extract information from a TempusServa system, and display it inline in other pages.

Example:

> Data extracted from the Tempus Serva connector
>
> http://alpha.tempusserva.dk/TempusServa/cmsinterface?q=examdates

> The final result inside a page in our website
>
> http://tempusserva.dk/site/index.php/da/eksamensdatoer

Notes on usage

- A CMS system is no requirement and static HTML files will work too.
- The TS backend will not be slowed because all content is cached in the connector (configurable)
- Connectors have NO influence on licensing

Tempus Serva setup (provider)

Frontend

1. Make the request that suits your needs
   - Filters and parameters
   - Sorting / grouping
   - Fields to display
   - Fields to display
   - Page size
2. Save the view
3. Click on the view and copy the parameters

Backend

1. Check that anonumous users have the right permissions
   - Note the interface will only allow READ operations
2. Go to "Integration" > "Content connector"
3. Add new element
   1. Give the connector a unique name (CONNECTOR_NAME>)
   2. Choose solution and set command type (as seen in the "command" parameter in the URL)
   3. Paste alle parameters from the URL above
      - Optionally add other settings like Language and Stylesheet
   4. Optionally define a variable that the "v" parameter will be mapped to (OPTIONAL_VARIABLE)
4. Test the new connector

URL format

- http://myserver.dk/TS/**cmsinterface?q=**<CONNECTOR_NAME>
- http://myserver.dk/TS/**cmsinterface?q=**<CONNECTOR_NAME>**&v=** <OPTIONAL_VARIABLE>

Example URL's

- http://myserver.dk/TS/**cmsinterface?q=customerList&v=3**
- http://myserver.dk/TS/**cmsinterface?q=longActivityList&v=3**
- http://myserver.dk/TS/**cmsinterface?q=singleActivityById&v=782382386**

Note that links between lists and single items will first be supported by Q2/2014.

CMS system setup (consumer)

Option: Client rendering

The following procedure

1. Make sure JQuery is available (normal JS can do the job)
2. Insert content placeholder and Javascript code

```
<script>
 jQuery.ajax(
 {
  url: 'http://myserver.com/TempusServaProxy.php?q=examdates',
```

```
    success: function(data) { jQuery('#tsContent').html(data); }
   });
</script>
```

Note: The URL above reflects the use of a proxy script (see below).

Option: Server side include

Insert code that fetches the content

```
echo file_get_contents("http://myserver.com/TS/cmsinterface?q=examdates")
```

In some cases you might want to remove the wrapper, header etc. from normal pages. This is done by adding the AjaxMode parameter.

```
echo
file_get_contents("https://talentpiper.com/demo/mainpublic?command=dk.p2e.blanket.codeunit.common.PagePublicRecordsListAndShow&AjaxMode=1");
```

Note: The URL above reflects the direct use of interface.

Overcoming CORS protection

I cases where the TempusServa server and the CMS system is on different domains (ex. acme.shared.com and cms.acme.com), browsers will prevent pages from accessing content from other servers.

Two options exist

1. Use server side includes (ok, but not supported everywhere)
2. Set up a mini proxyserver

A mini proxyserver written i PHP is very simple (aspx/jsp will have similar features) and placed on the CMS side.

```
<?php
echo file_get_contents("http://myserver.dk/TS/cmsinterface?q=".$_GET["q"]."&v=".$_GET["v"]);
```

The proxy can also be deployed on another domain than the CMS system.

```
<?php
header("Access-Control-Allow-Origin: *");
echo file_get_contents("http://myserver.dk/TS/cmsinterface?q=".$_GET["q"]."&v=".$_GET["v"]);
```

For a more safe version write the name of the CMS domain

```
header("Access-Control-Allow-Origin: www.tempusserva.dk");
```

Afterwards you just make the calls through the proxy using exact same parameters.