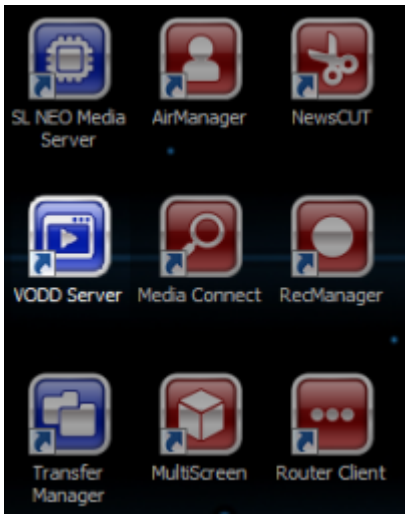
SL NEO VOD

The Skylark VOD service allows creating a VoD (Video on Demand) Internet portal that implements the system of individual delivery of TV programs and movies from the Skylark media-server to the subscriber in different formats.



Launching the VOD server is implemented by the VODD Server icon. As VODD interacts with the database, SL NEO Media Server with a configured database service must be launched beforehand.



Launched VODD Server places its icon near the clock.

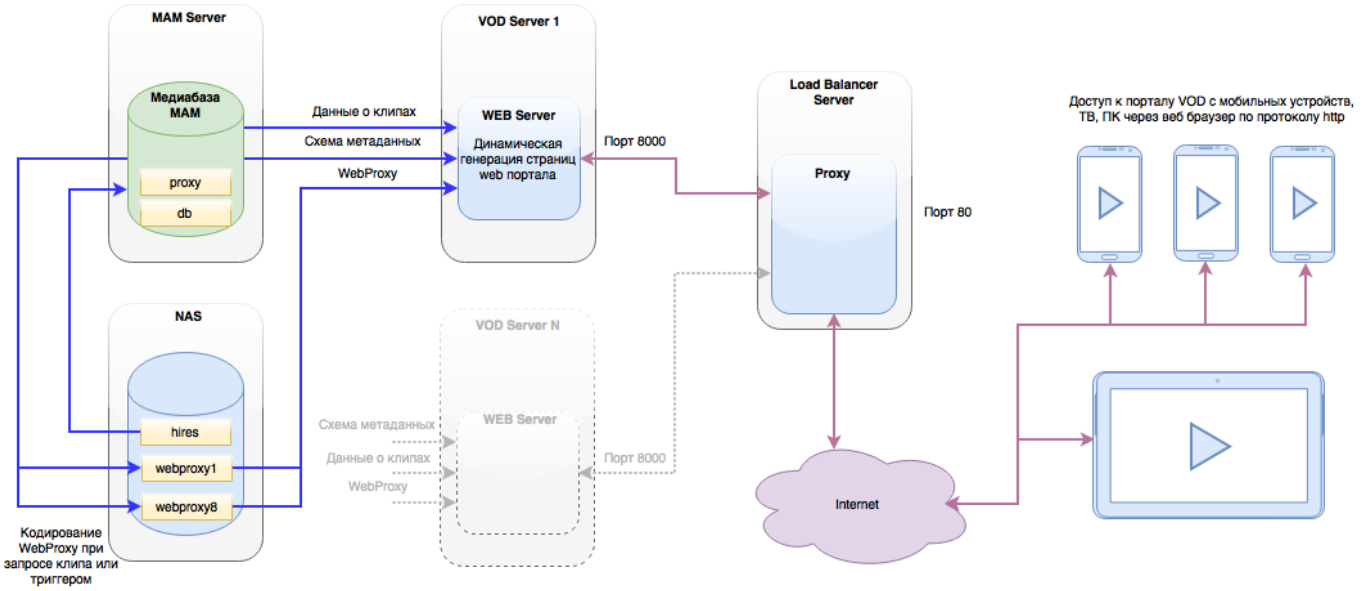
The information is relevant for Skylark SLNeo 2.1 software and higher.

VOD Basic Components

- VOD Service (vodd.exe) – creating a web-portal available via the 8000 port.
- Database service (stored.exe) – storing materials published in the VOD portal, WebProxy access.
- Media Connect client application (maconnect.exe) – manual import, managing and describing clips placed in VOD.
- Transfer Manager server application (transfermgr.exe) – automated transfer and processing materials in the VOD server database.

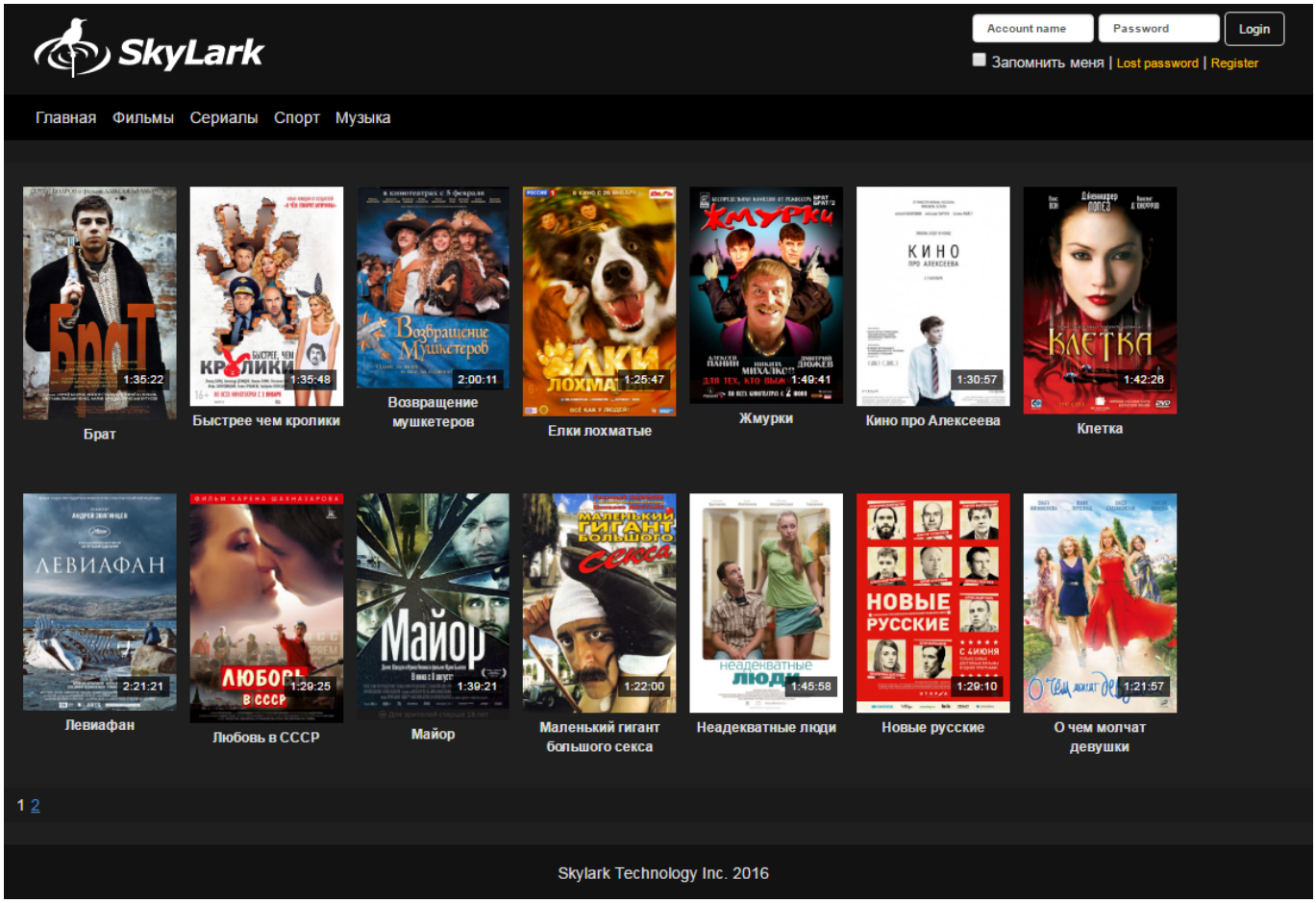
VOD Architecture

The picture displays one of variants of implementing VOD on the SL NEO Platform.

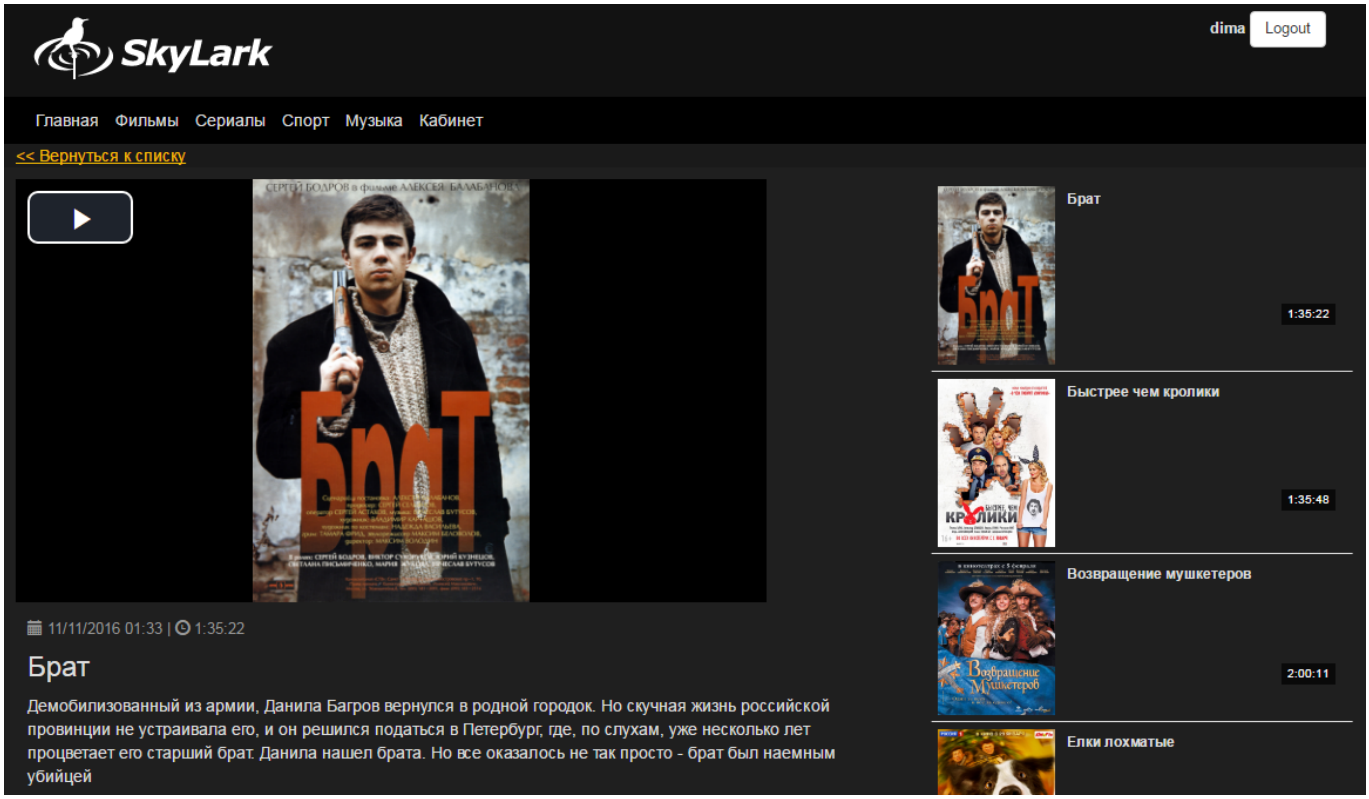


Portal

The portal can be flexibly customized for certain client tasks, integrated with social media, an existing subscriber base and payment systems.



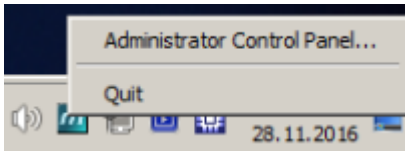
The list of movies.



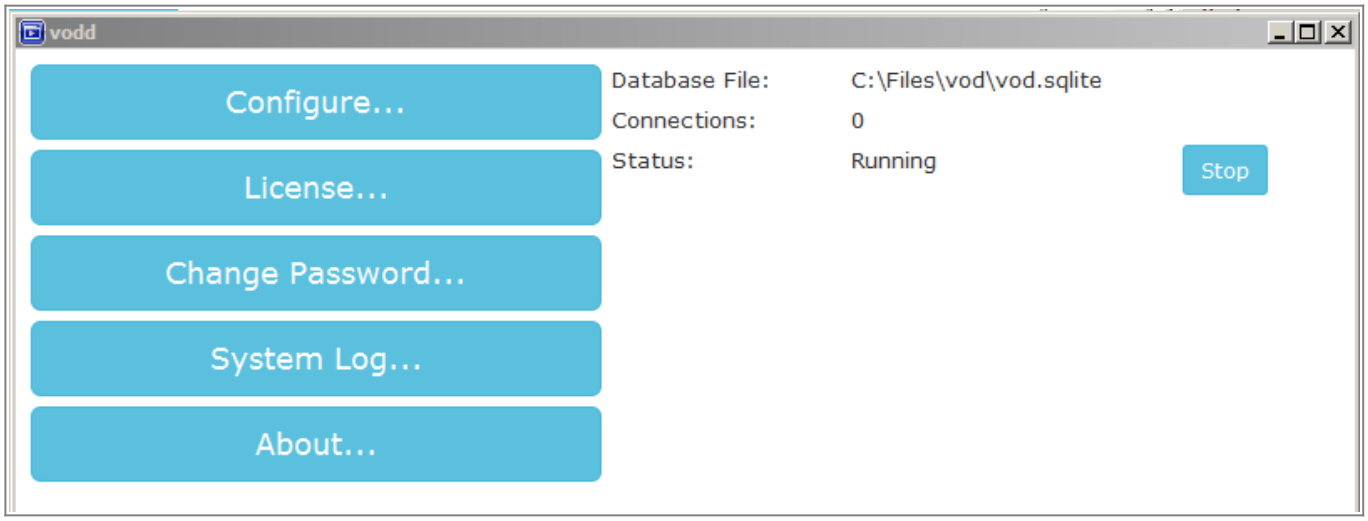
An example of a movie page available after login.

Vodd Control Panel

All settings implemented in the control panel are saved to the file: C:\Program Files (x86)\SL NEO Media Platform\run\vodd_config.xml.



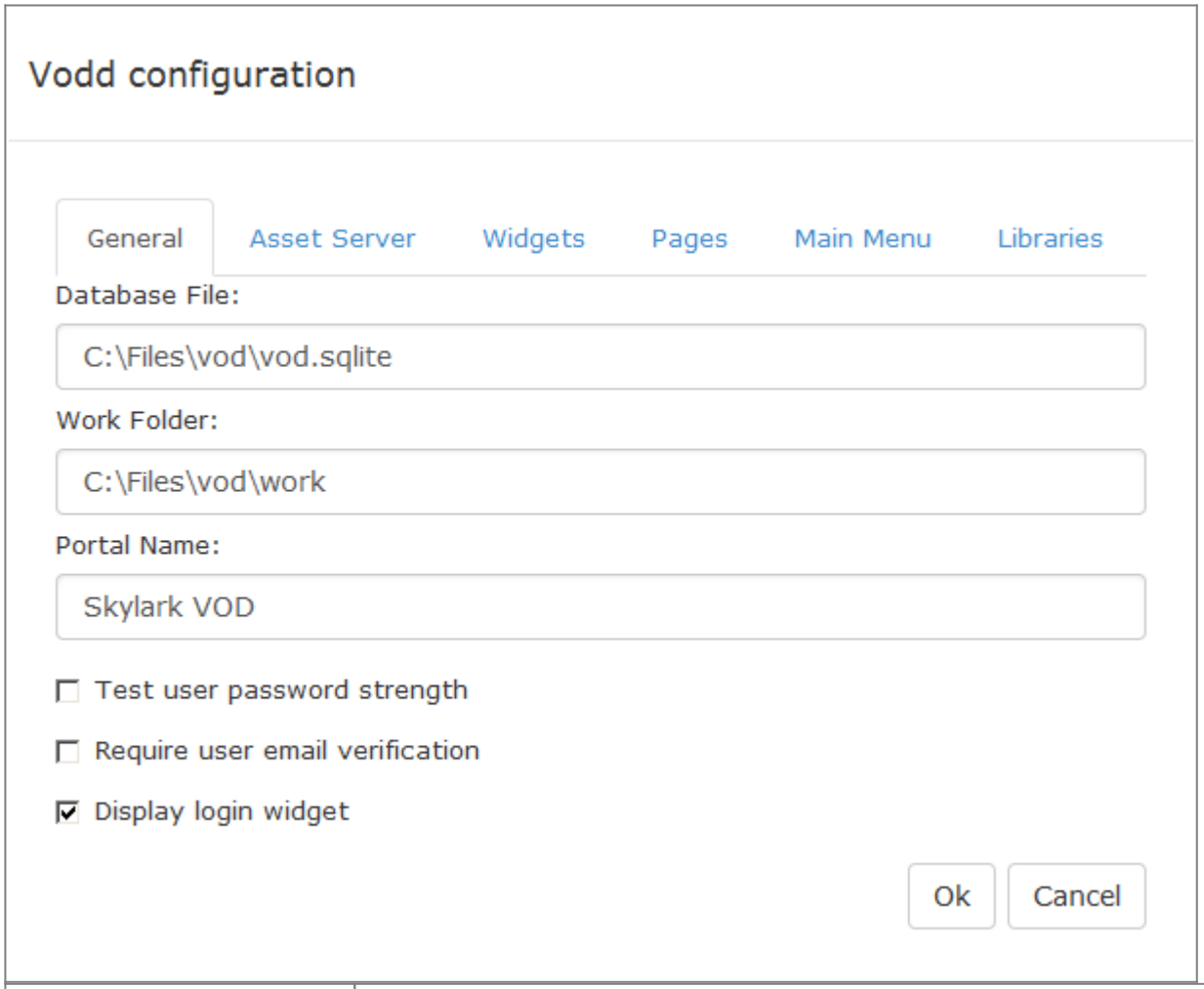
Login to the Administrator Control Panel is implemented by the vodd service icon.



Parameter	Value
Database File	The SQLite database path. The database stores credentials of VOD portal users and is created automatically, during the server first launch.
Connections	The number of active connections to the portal web-interface.
Status	The portal web-interface status. Running – launched and available to users, Stopped – the app is unavailable, users get the “Server is under maintenance” message.
Start/Stop	The button to launch and stop the portal web-server.
Configure	The menu to access VOD portal settings.
License	License information.
Change Password	Changing the portal admin password.
System Log	Displaying the log of current user activity in the real-time mode.
About	Information on the VOD server version.

VODD configuration

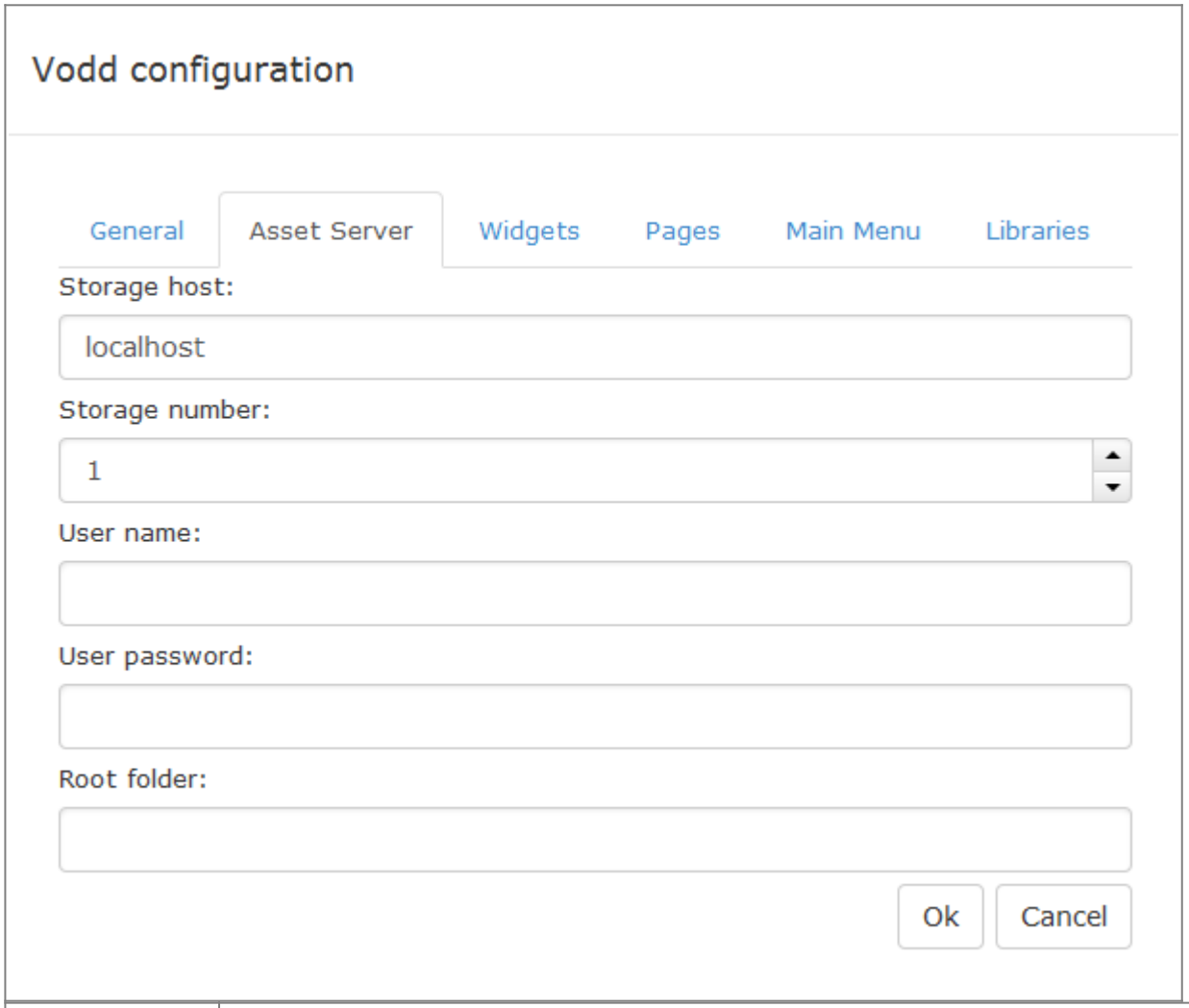
General



Parameter	Value
Database File	Path to the VOD database file, containing information on registered users. By default: `C:\Files\vod\vod.sqlite`.
Work Folder	Path to the folder for temporary files.
Portal Name	The portal name that will be used in the <TITLE></TITLE> tag.

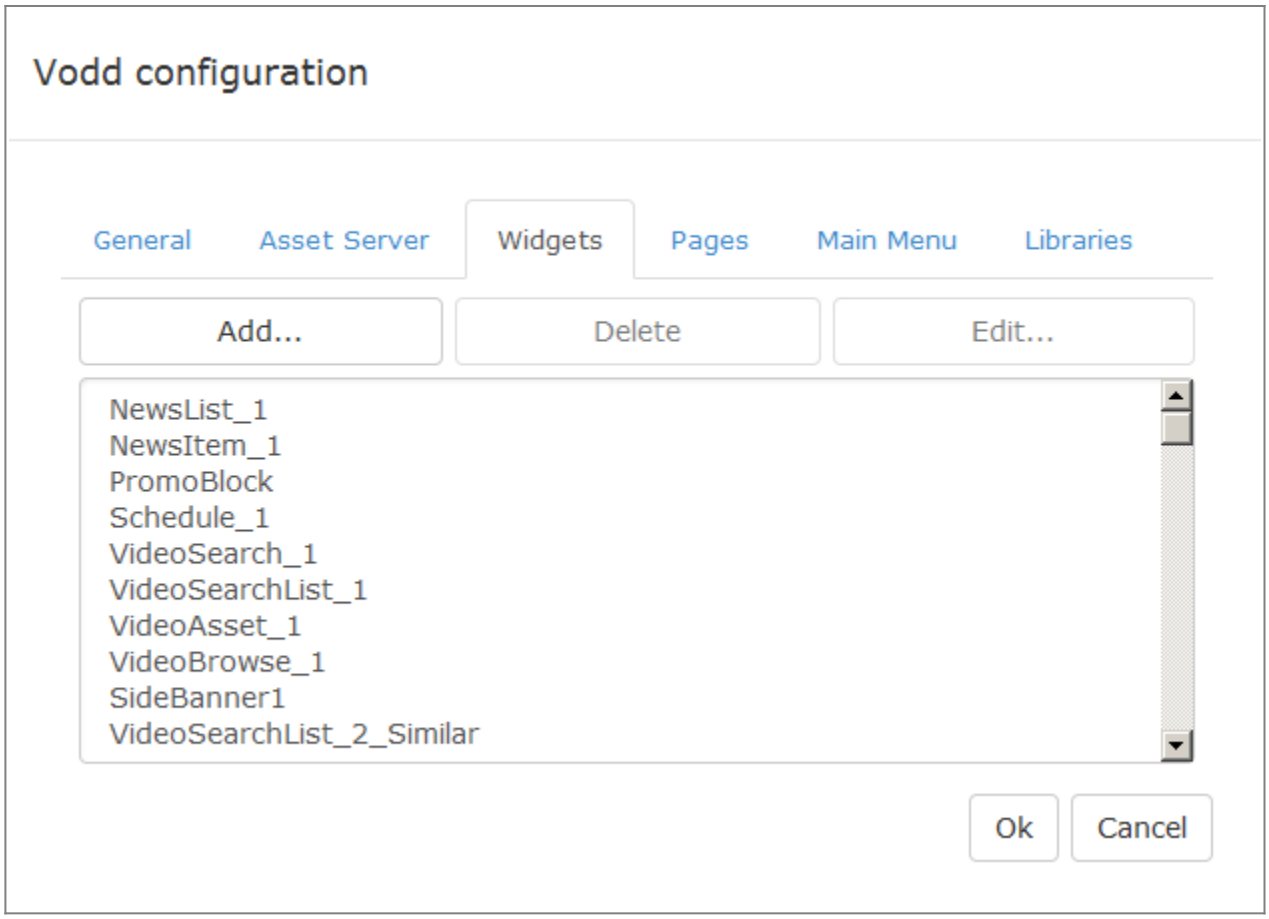
Parameter	Value
Test user password strength	Setting requirements to the password complexity during registration.
Require user email verification	Activating email verification during registration.
Display login widget	Displaying a login widget at the top of portal pages.

Asset Server



Parameter	Value
Storage host	The IP-address of a server with the configured database. The default value is localhost.
Storage number	The database service sequence number on the server with the specified IP address.
User name	The username for the database access.
User password	The user password for the database access.
Root folder	The database root folder, relative to which newsitem and videoasset would be selected. The default value (empty) is the database root. The path is specified in the “folder1/folder2” format.

Widgets

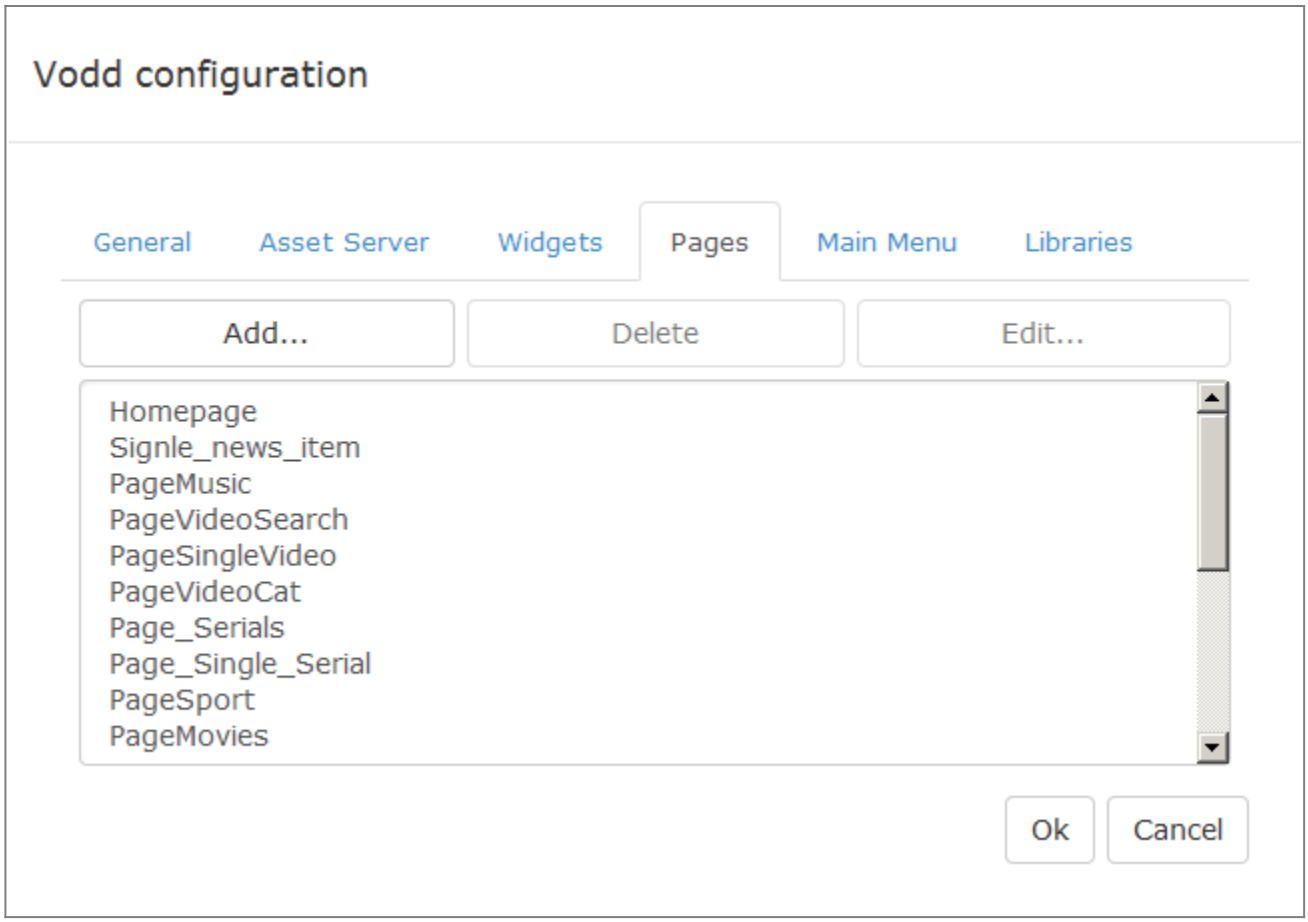


A widget is a functional element of the page, responsible for data input and output and able to interact with other widgets.

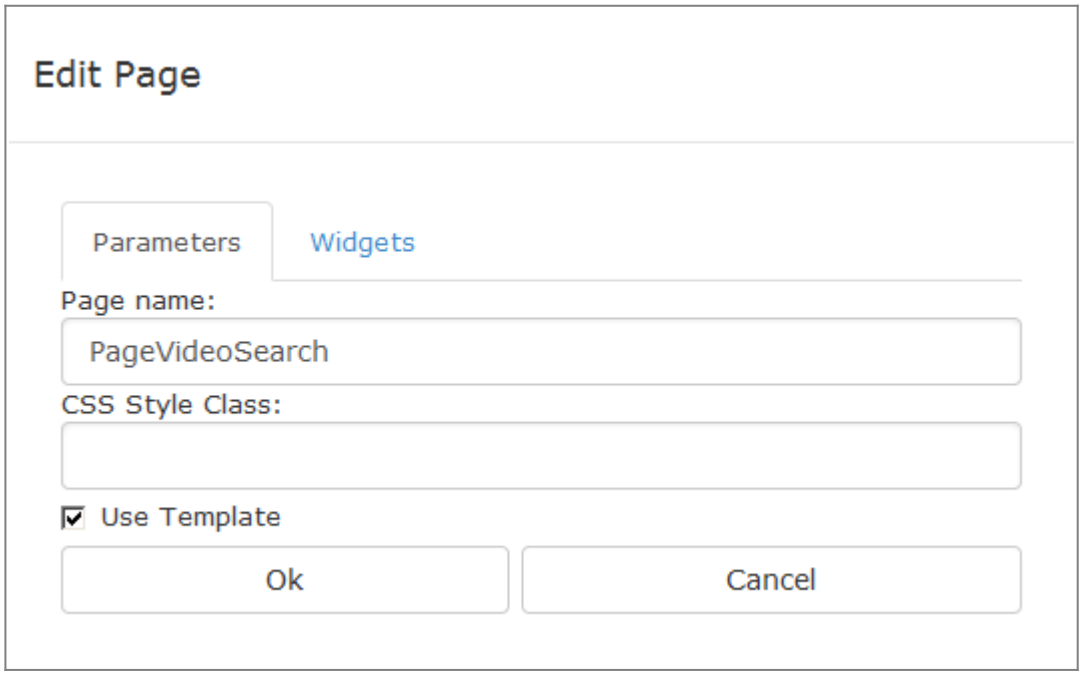
Parameter	**Value**
Add	Adding a new widget.
Delete	Removing the selected widget.
Edit	Activating the editing window for a selected widget.

Pages

A page is the portal element that unites a group of widgets and has a design template.



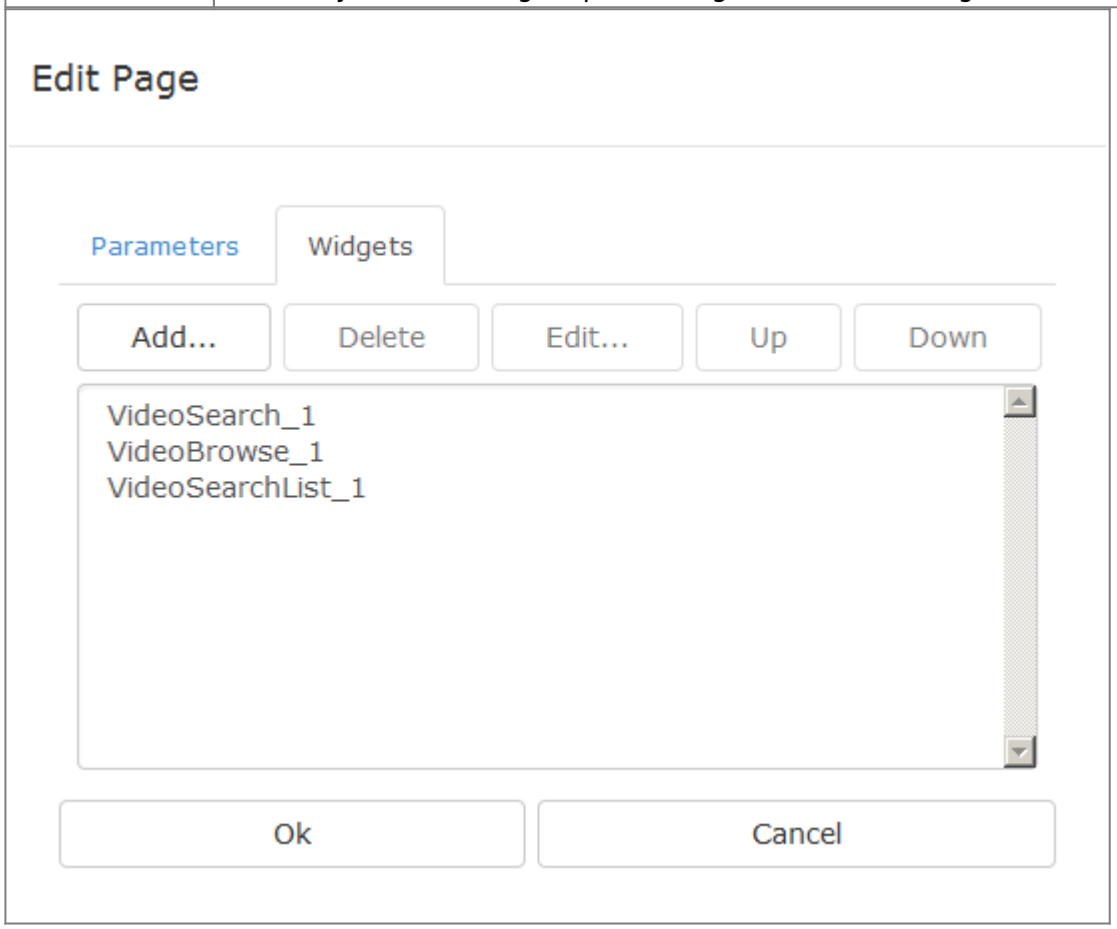
The list of pages created in the portal.



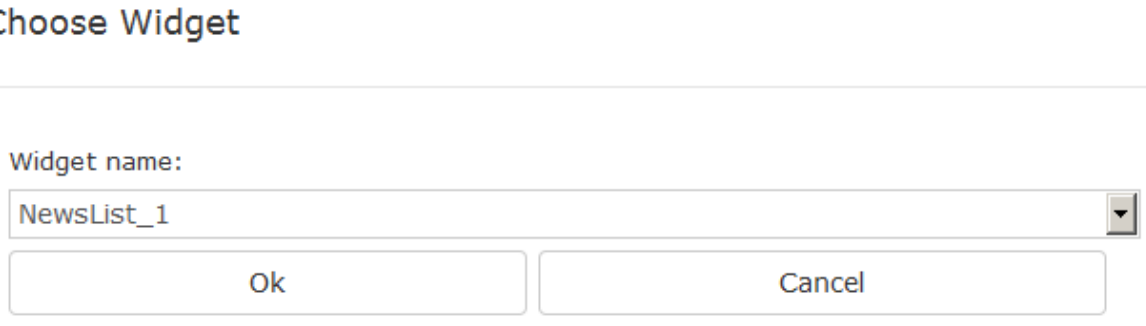
Parameters

Parameter	**Value**
Page name	The page name used for widget interaction.
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.

Parameter	Value
Use Template	When the option is active, the vodd.user.page-NAME template is used for displaying the page (NAME is the page name). Widgets will be anchored to their names in the page template. All templates are stored in the file: C:\Program Files (x86)\SL NEO Media Platform\docroot\vodd\user.xml, in <message id="vodd.user.page-NAME"> tags. If the option is not used, or a template is not found, consecutive widget output is used by default (widgets positioning is set on the Widgets tab).

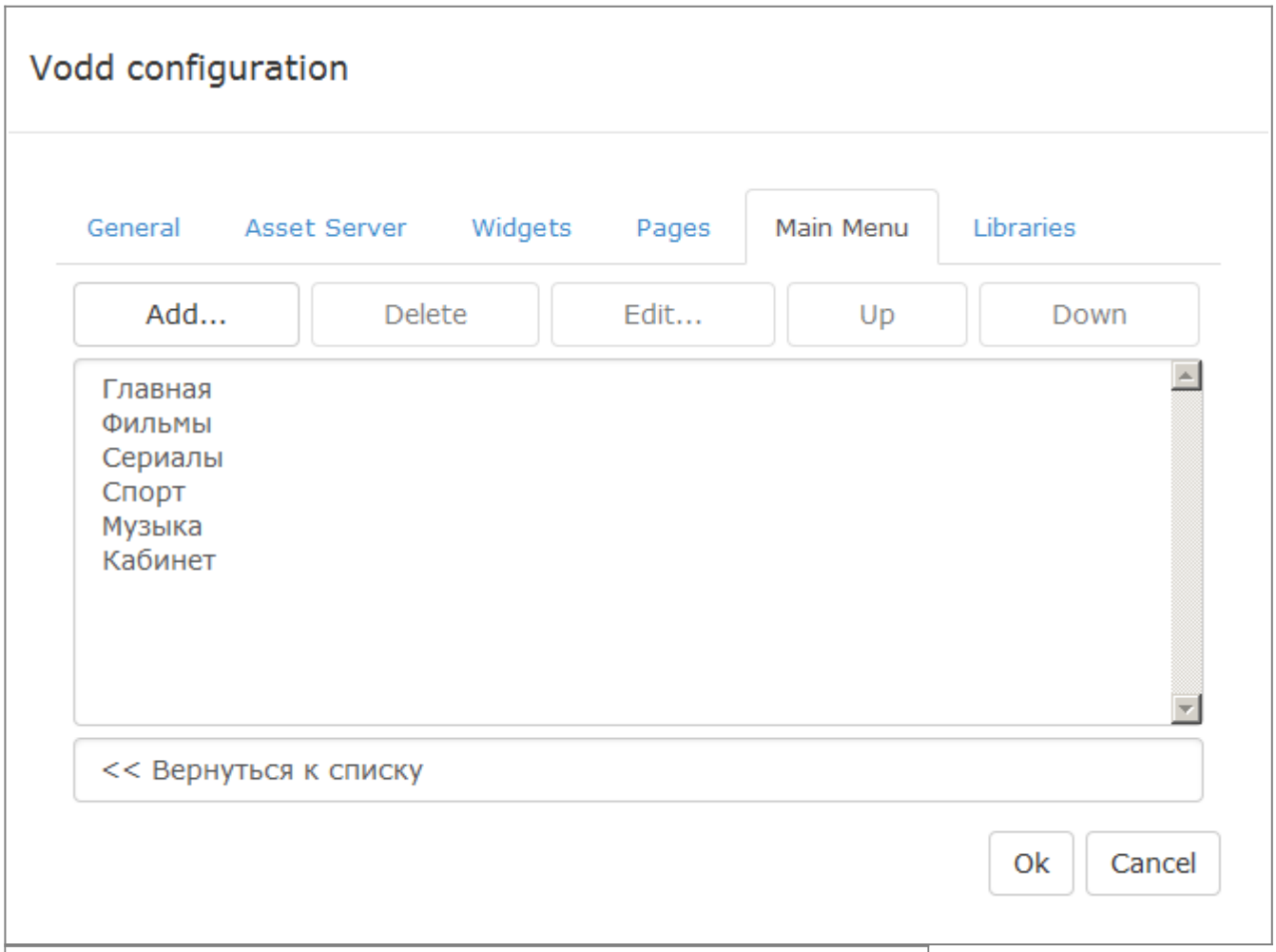


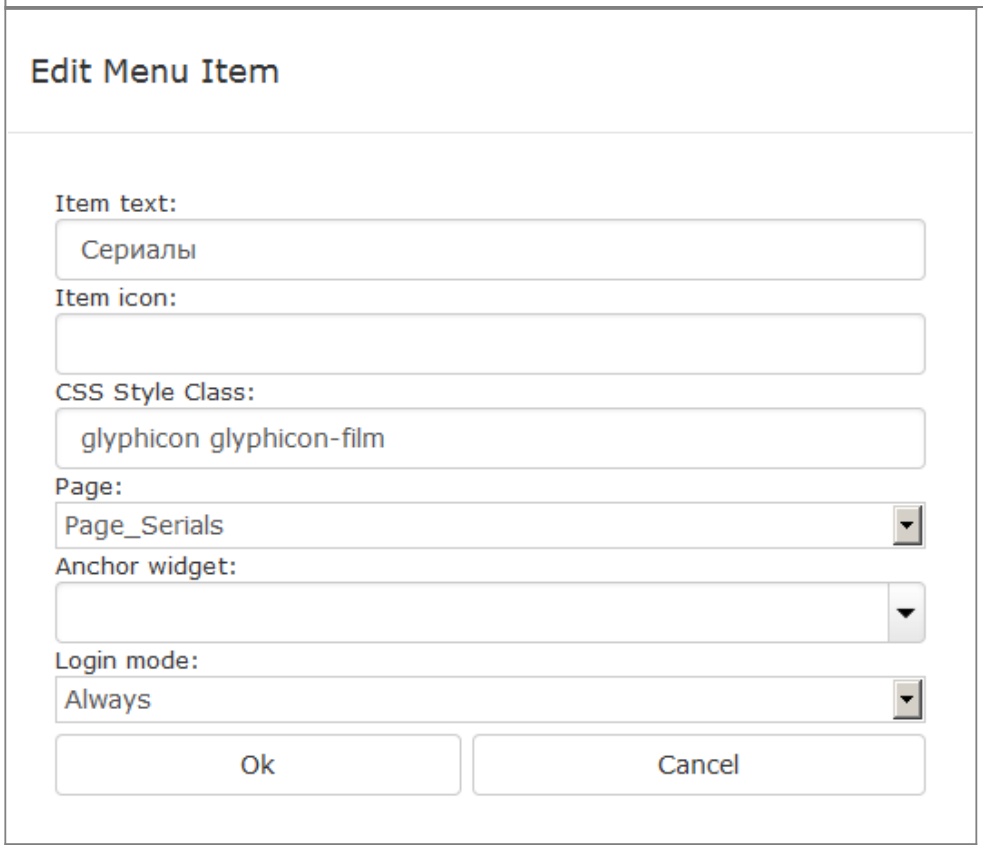
Widgets

Parameter	Value
Add	Selecting a widget for adding to the page 
Delete	Removing the selected widget.
Edit	The editing dialogue for replacing the selected widget.
Up	Moves the selected widget up.
Down	Moves the selected widget down.

Main Menu

Configuring the portal main menu.





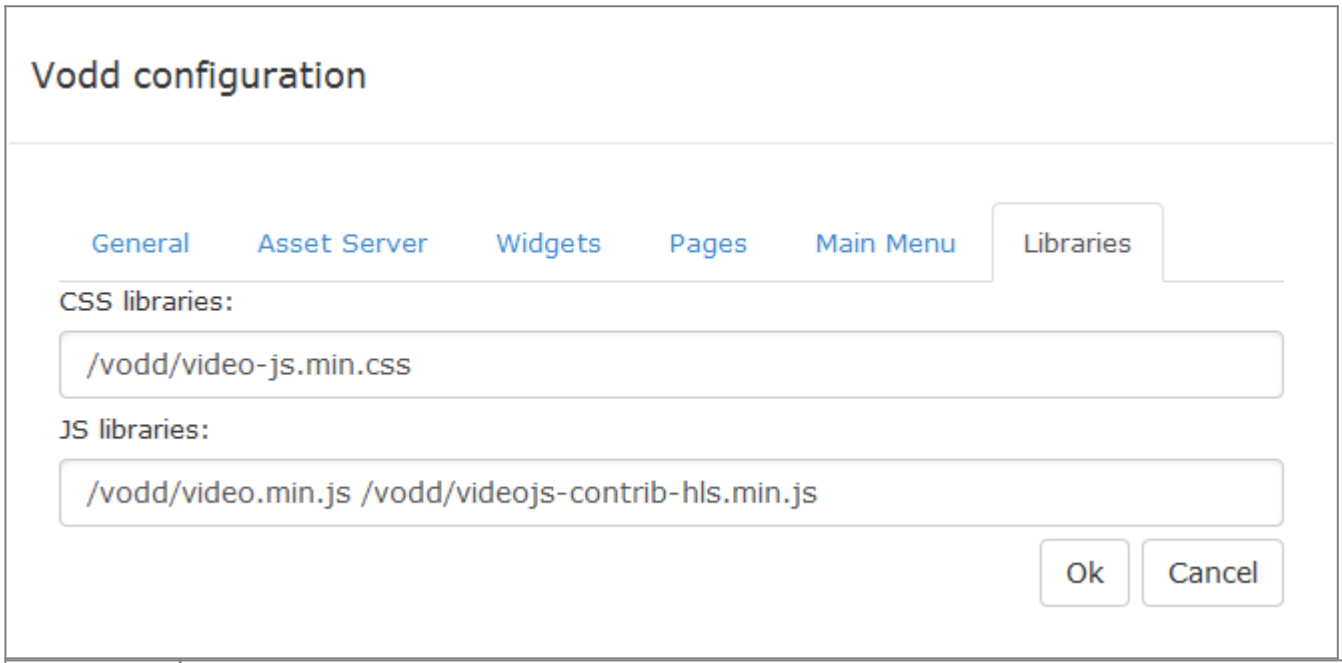
Edit menu Item

Parameter	Value
Item Text	The menu item title.
Item Icon	The menu item icon. If the picture is taken from the database, specify the virtual path `images/FILE_NAME`, or the name of the file with extension, if it is placed in the folder `C:\Program files (x86)\SL NEO Media Platform\docroot\`
CSS Style Class	Setting a CSS class for a certain menu item.
Page	Selecting a menu item landing page from the list.
Anchor widget	Scrolling the page to the required widget, when going to the page.
Login mode	Setting visibility of a menu item, depending on the user authorization status in the portal, when going to the page. • Always – the menu item is always visible. • LoggedInOnly – the menu item is visible, if the user is authorized. • LoggedOutOnly – the menu item is visible, if the user is not authorized.

No name input field – specifying the text showed when displaying VideoAsset and NewsItem widgets, to be able to return to VideoSearchList and NewsList respectively. For example: “Return to the list”.

The “main_menu_item_current” css class is added for main menu elements in case, if the menu item becomes current.

Libraries

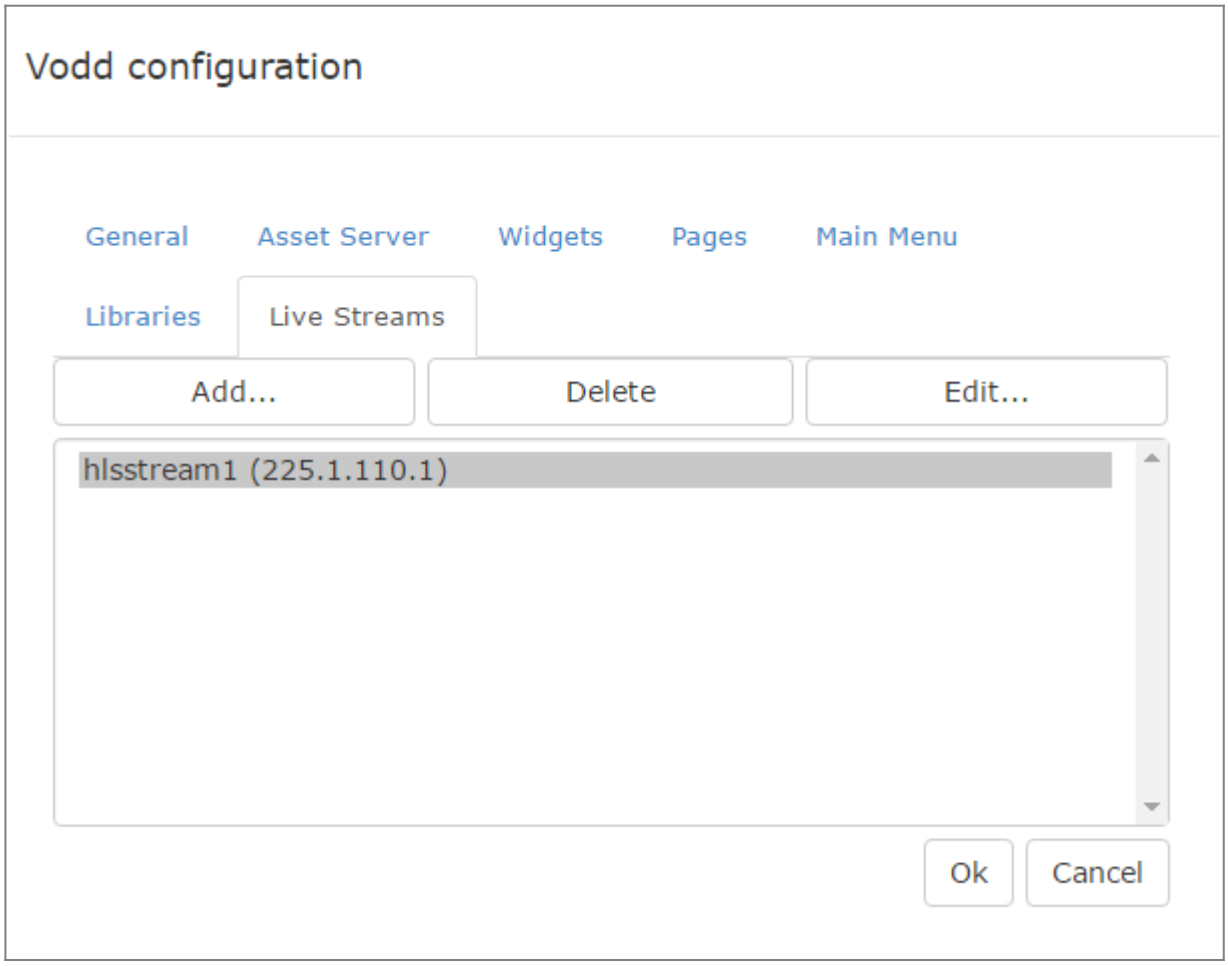


Parameter	Value
CSS libraries	Paths to CSS files are specified space-separated. Files will be connected in the beginning of the <BODY> section, in the <link href=“…” rel=“stylesheet” type=“text/css”> tag that points to the virtual resource dynamically formed by the server. All external CSS files in this resource are specified via @import. Example: `vodd/video-js.css`. Files must be placed in the catalogue: `C:\Program Files (x86)\SL NEO Media Platform\docroot\vodd\`.

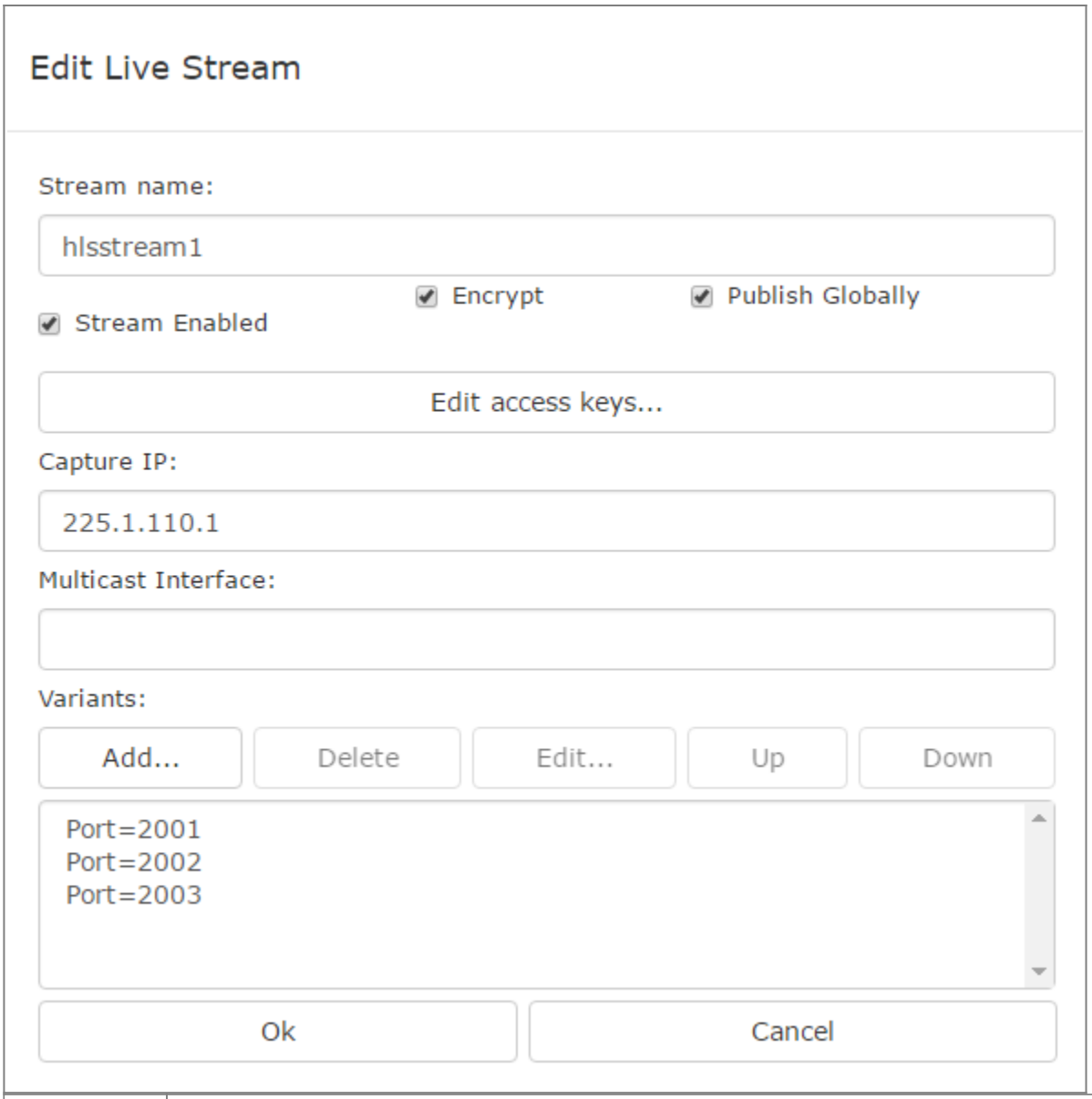
Parameter	Value
JS libraries	Paths to JS files are specified space-separated. Files will be connected in the <HEAD></HEAD> section of the portal. Example: `vodd/video.js`. Files must be placed in the catalogue: `C:\Program Files (x86)\SL NEO Media Platform\docroot\vodd\`

Live Streams

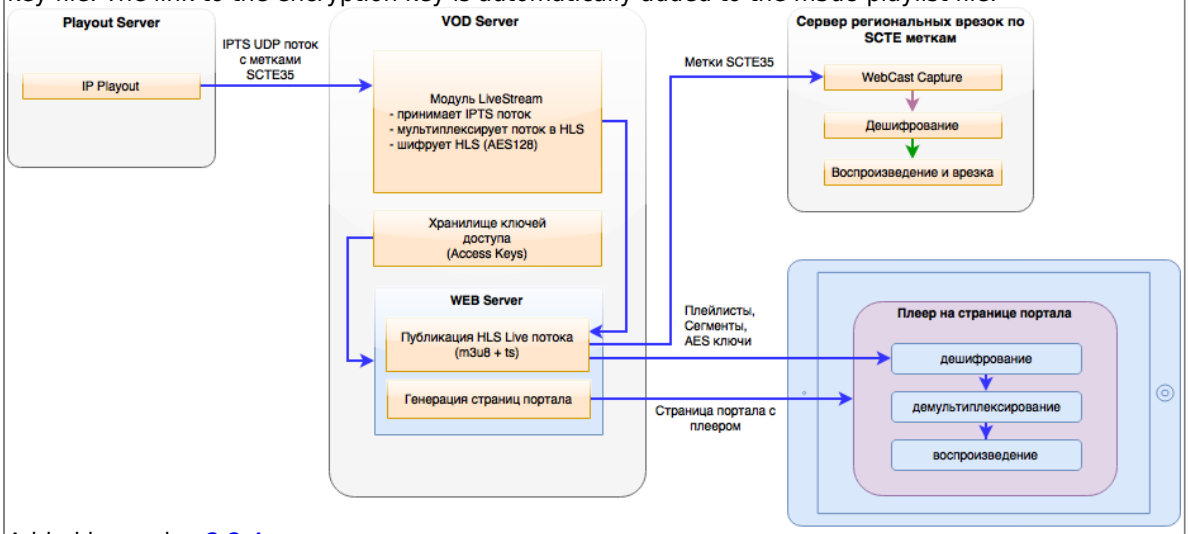
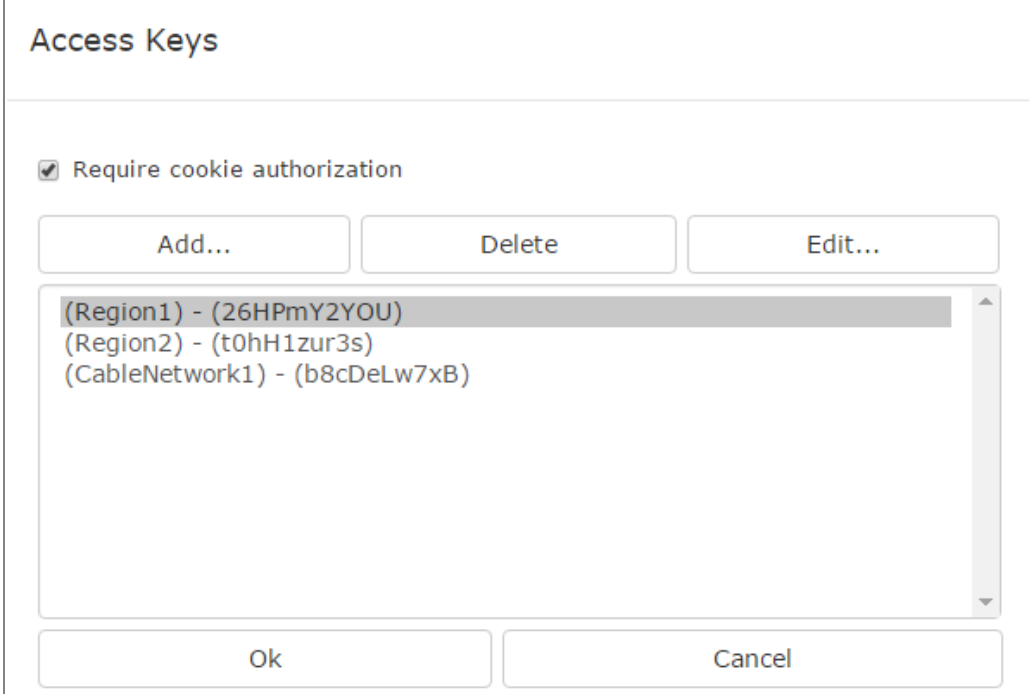
The Live Streams tab allows configuring capture of multi-bitrate IPTS-streams for their rebroadcasting to HLS streams.

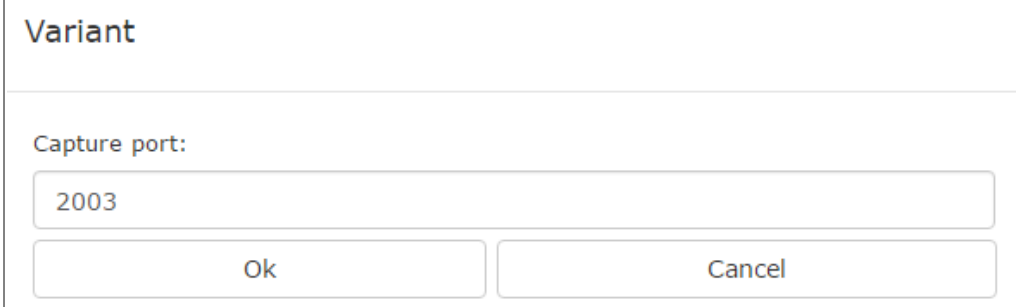


The window for adding a new stream.



Parameter	Value
Stream name	The stream name, which is simultaneously the name of the HLS playlist file, and is used for access to the stream from widgets and via URL. If access to the stream will be implemented via URL from a browser, it is recommended to specify an *.m3u8 extension.
Stream EnabledThe option activates rebroadcasting streams in the HLS format.	

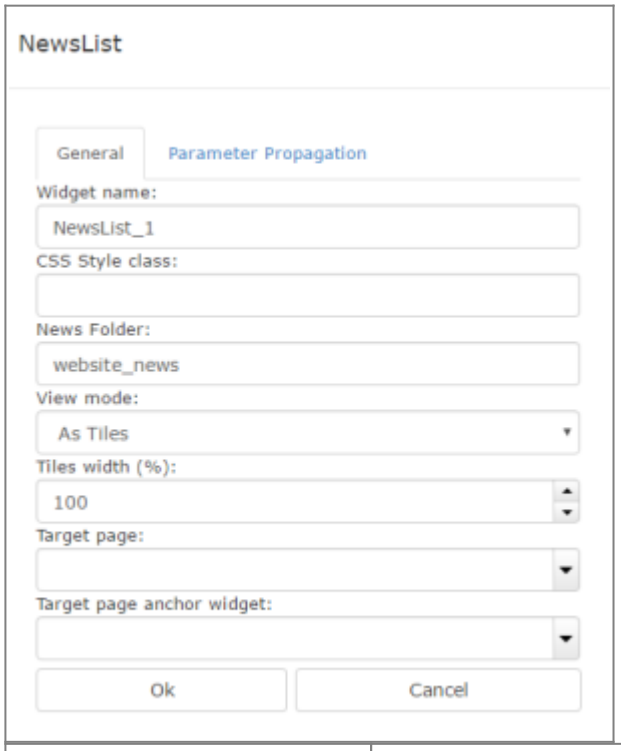
Parameter	Value
Encrypt	Activating AES128 data encryption of the HLS stream. Viewing such a stream requires the encryption key file. The link to the encryption key is automatically added to the m3u8 playlist file.  Added in version 2.2.4.
Edit access keys...	Managing the storage of access keys to the AES encryption key – Access keys. The storage allows creating keys that are random “Key + Key value” pairs used for access to the encrypted HLS stream without actual login to the VOD portal.  The option **Require cookie authorization** activates the client authorization check through Cookie before sending him an AES encryption key. To successfully pass the check, the user must send: • The session cookie, automatically generated and sent through the portal in case of an encrypted stream request. The mechanism works transparently for all portal users and does not require any additional actions. • Cookie Name and Cookie Value, specified in the Capture service settings. These values must be previously saved in the Access Keys storage. This option is used, for instance, for regional commercial inserts, when the encrypted HLS stream is requested directly, not using the portal. If the **Require cookie authorization** option is inactive, the AES key file and viewing the encrypted HLS stream are available to all clients. Added in 2.2.4 version.

Parameter	Value
Publish Globally	The option grants access to the Live HLS stream via a direct URL address, with no necessity of access through the web-interface. The stream is published by the address: http://IP_ADDR:8000/live/NAME, where NAME is the Live HLS name specified in the configuration. The option can be used for regional commercial inserts, to capture a stream via IP Capture in the WebCast mode.
Capture IP	The IP address to be used for capturing IP streams. For example: 225.1.110.1.
Multicast Interface	The IP address of the interface to be used for capturing IP streams.
Variants	 The option allows specifying port numbers for the set IP address, which provide stream versions with different bitrate. This allows creating a playlist that supports adaptive bitrate.

SCTE35 markers from the input IPTS stream are sent to the output HLS stream (to the TS, but not exported to the m3u8 playlist yet).

Widget Types

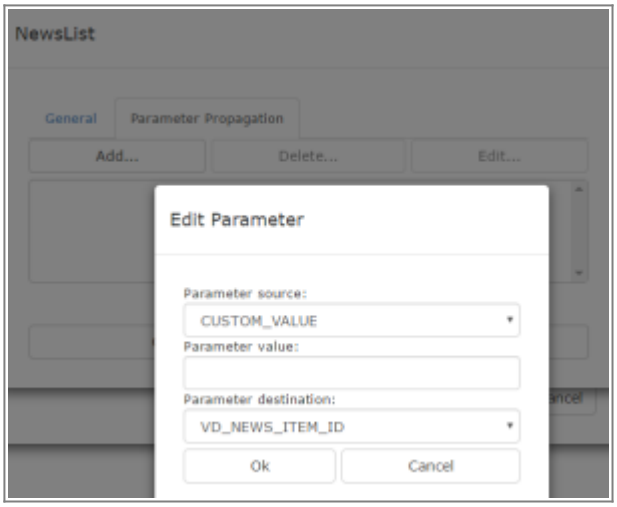
NewsList



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in <DIV class=field value></DIV>.

Parameter	Value
News Folder	The folder for reading off the Note type elements.
View mode	THe list displaying type: • As List - a consequent list • As Tiles - tiles
Sort direction	The sorting order for the As List displaying mode: • Ascending - direct output • Descending - reverse output
Tiles width (%)	The widget width relative to the parent element, for the As Tiles displaying mode.
Target page	The landing page for a Note element; it must contain one or several NewsItem elements.
Target page anchor widget	Selecting a NewsItem element that will display the Note item data.

The Parameter Propagation tab.



The setting allows sending values to the landing page variables. Variables are available to all page widgets.

Parameter	Value
Parameter source	The transferred value type: • CUSTOM_VALUE – a random string value will be sent. • PASS_THROUGH – sending the current value selected in Parameter value. • XMP_FIELD - sending the XMP field value for the selected clip (specified in the `namespace:field_name` format). Example: `sl:title`).
Parameter value	The value of the transferred parameter, depending on the Parameter source field.

Parameter	Value
Parameter destination	The landing page variable that will store the transferred value. • VD_NEWS_ITEM_ID • VD_VIDEO_SEARCH_REQUEST • VD_VIDEO_SEARCH_REQUEST_FIELDS • VD_VIDEO_SEARCH_REQUEST_FOLDER • VD_VIDEO_SEARCH_FILTER • VD_VIDEO_ITEM_ID

When displaying portal pages and updating page parameters, values of parameters are displayed in the System Log of the control panel.

NewsItem

The element to output a piece of news.



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.
Default News Item ID	The Note element MediaID that will be output by the NewsItem element.

TextBlock

The widget that allows to output an arbitrary HTML code.



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in <DIV class=field value></DIV>.
XHTML Text	Using JavaScript is not allowed.

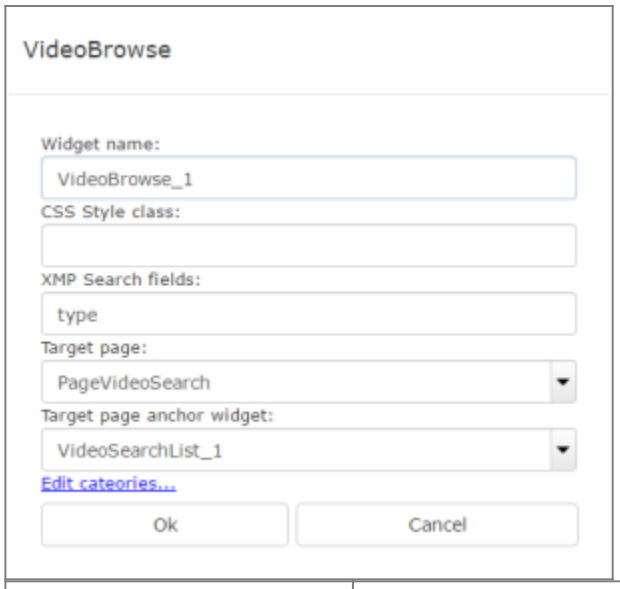
VideoSearch

The widget displaying the entry field of the search request for VideoSearchList.

Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in <DIV class=field value></DIV>.
Button Enabled	Displaying the search button. ON - enable the search button, OFF - disable the search button (search is implemented by pressing Enter).
Button Text	Text displayed at the search button.
Empty Text	Text displayed in the empty search entry field.
Target page	The search request landing page name.
Target page anchor widget	The VideoSearchList widget at the landing page, receiving the search request.

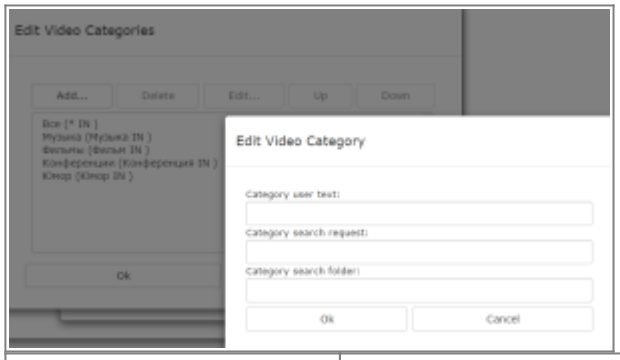
VideoBrowse

Displaying a preliminarily configured list of search requests, connected with certain XMP-fields.



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.
XMP Search fields	The metadata field list for the search. Specify the list of fields comma-separated, or leave the value empty for searching by all fields. For example: `dc:title, dc:description, dc:place` or `sl:keywords`
Target page	The list element landing page name.
Target page anchor widget	The VideoSearchList widget at the landing page, receiving the search request.
Edit categories	Editing the list of search requests.

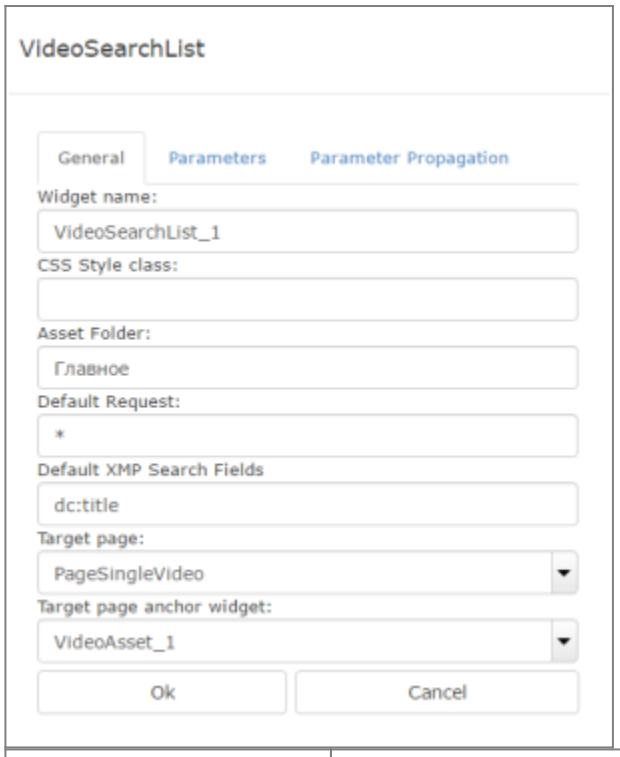
Edit categories.



Parameter	Value
Category user text	The value to be displayed in the portal interface.
Category search request	The value to be sent as a search request.
Category search folder	The path to the folder for searching, specified relative to the VOD root directory, e.g.: `films/new`.

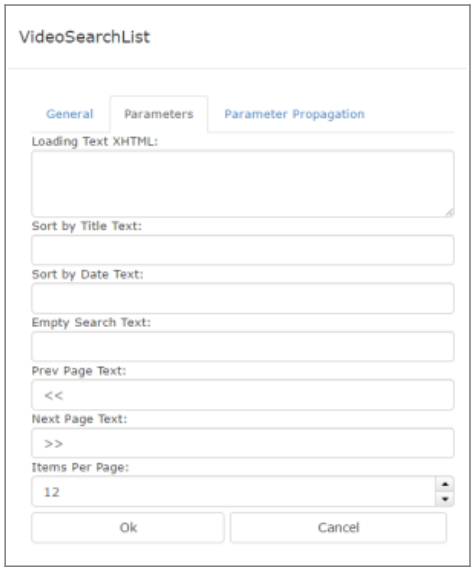
VideoSearchList

The feature displays the list of video-clips according to the received parameters. It receives search parameters from VideoSearch and VideoBrowse widgets. The General tab.



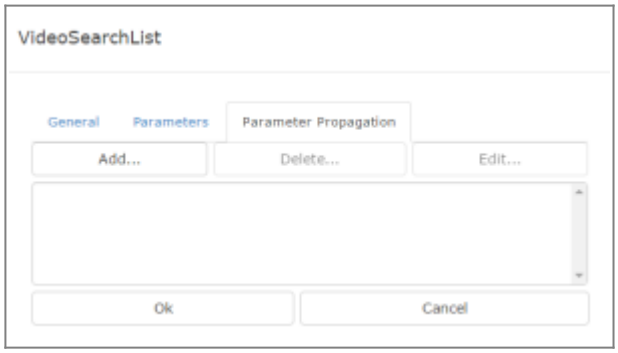
Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.
Asset Folder	The path to the folder for the search, specified relative to the VOD root directory, e.g.: `films/new`.
Default Request	The default request used if the widget receives an empty value. “*” – selecting all values.
Default XMP Search Fields	The list of metadata fields, in which the search will be implemented. Specify the list of fields comma-separated, or leave the value empty for searching by all fields. For example: `dc:title, dc:description, dc:place` or `sl:keywords`
Target page	The list element landing page name.
Target page anchor widget	The VideoSearchList widget at the landing page, receiving the selected clip ID.

The Parameters tab.



Parameter	Value
Loading Text XHTML	Fix Me!
Sort by Title Text	Fix Me!
Sort by Date Text	Fix Me!
Empty Search Text	Fix Me!
Prev Page Text	The text displayed at the button for moving to the previous search result page.
Next Page Text	The text displayed at the button for moving to the next search result page.
Items Per Page	The number of found elements displayed at one page.

The Parameter Propagation tab.

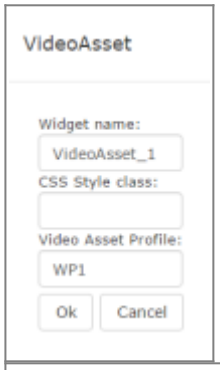


The setting allows sending values to the landing page variables. Variables are available to all page widgets.

Parameter	Value
Parameter source	The transferred value type: • CUSTOM_VALUE – a random string value will be sent. • PASS_THROUGH – sending the current value selected in Parameter value. • XMP_FIELD – sending the XMP field value for the selected clip (specified in the `namespace:field_name` format), for example: `sl:title`.
Parameter value	The value of the transferred parameter, depending on the Parameter source field.
Parameter destination	The landing page variable that will store the transferred value. • VD_NEWS_ITEM_ID • VD_VIDEO_SEARCH_REQUEST • VD_VIDEO_SEARCH_REQUEST_FIELDS • VD_VIDEO_SEARCH_REQUEST_FOLDER • VD_VIDEO_SEARCH_FILTER • VD_VIDEO_ITEM_ID

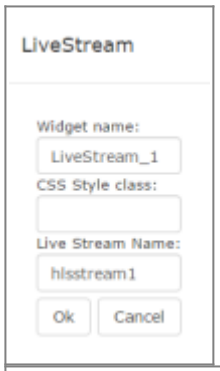
When displaying portal pages and updating page parameters, values of parameters are displayed in the System Log of the control panel.

VideoAsset



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.
Video Asset Profile	The profile name to be used for clip playback: WP1-WP8. For example, `WP1`.

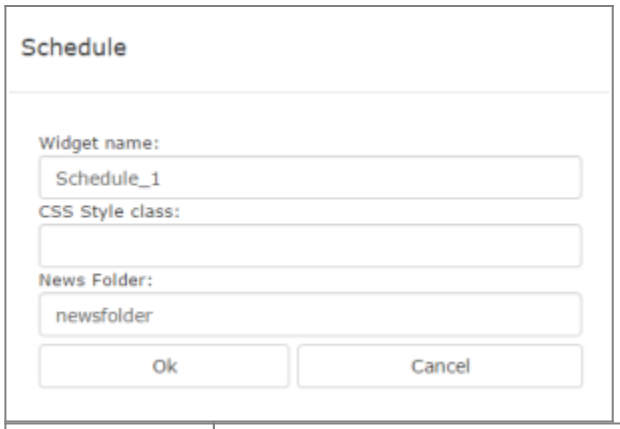
LiveStream



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in `<DIV class=field value></DIV>`.
Live Stream Name	The name of the stream configured at the Live Streams tab, which will be played by the widget.

Schedule

The widget implements playback of VOD content by playlists. Added in 2.2.5 version.



Parameter	Value
Widget Name	The widget name
CSS Style class	The widget output HTML will be enclosed in <DIV class=field value></DIV>.
News Folder	

```
<message id="vodd.user.root-container">
    ...html tags and built-in variables...
</message>
```

Page Templates

If the Use Template option is selected in page settings, the vodd.user.page-NAME'' template (where NAME is the page name) will be used for displaying the page. If the option is not used, or a template is not found, consecutive widget output is used by default. <code> <message id="vodd.user.page-NAME"> …html tags and built-in variables… </message> </code> ==== Widget Templates ==== The vodd.user.NAME'' template (where NAME is the widget identifier) is used for displaying widgets.

```
<message id="vodd.user.NAME">
    ...html tags and built-in variables...
</message>
```

The table of available widget identifiers.

NAME	Widget Name	Note
news-item	NewsItem	
news-list-item	NewsList	
video-asset	VideoAsset	
video-search-item	VideoSearchList	
video-search	VideoSearch	
logged-in	-	The system widget of an authorised user.
login	-	The system widget of the authorisation form.

Template Variables

Variable	Description
${user-name}	
${password}	

Variable	Description
Fix Me!	

The following template variables that can be parametrized by simple strings (and can be used, for example, as “src” for the “img” tag) has been added:

vodd.user.news-item: news_item_picture_url vodd.user.news-list-item: news_list_item_icon_url vodd.user.video-search-list-item: video_search_item_icon_url



Further work required.

Scripts in Templates

JavaScript can be used in templates in the following way:

```
<script type="text/javascript">
      // <![CDATA[

      ... javascript code ...

      // ]]>
    </script>
```

user.xml Example

The default template example: ''C:\Program Files (x86)\SL NEO Media Platform\docroot\vodd\user-default.xml''.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<messages xmlns:if="Wt.WTemplate.conditions">

  <message id="vodd.user.login">
    <label class="form-horizontal">${user-name}</label>
    <label class="form-horizontal">${password}</label>
    ${remember-me} Remember me
    ${login class="btn-primary"}
    ${lost-password class="btn-link"} ${sep} ${register class="btn-link"}
    ${<if:oauth>}, or use ${icons}${</if:oauth>}
  </message>

  <message id="vodd.user.logged-in">
    <b>${user-name}</b> ${logout}
  </message>
```

```
<message id="vodd.user.root-container">
  ${back_button}
  ${menu_bar}
  ${login_widget}
  ${page_container}
</message>

<message id="vodd.user.page-not-found">
  ${label}
</message>

<message id="vodd.user.video-search">
  ${request_edit} ${search_btn}
</message>

<message id="vodd.user.video-search-list-item">
  ${video_search_item_icon}
  ${video_search_item_title}
  ${video_search_item_duration}
  ${video_search_item_summary}
  ${video_search_item_date}
</message>

<message id="vodd.user.video-asset">
  <video poster="${video_asset_poster_url}"
         controls="true"
         preload="none"
         class="video-js vjs-default-skin">
    <source src="${video_asset_url}"
            type="${video_asset_media_type}"/>
  </video>

  ${video_asset_title}
  ${video_asset_summary}
  ${video_asset_text}
  ${video_asset_duration}
  ${video_asset_datetime}

  <script type="text/javascript">
    // <![CDATA[

    var video = this_widget.getElementsByTagName('video')[0];
    videojs(video, {}, function() {});

    // ]]>
  </script>
</message>

<message id="vodd.user.news-list-item">
  ${news_list_item_icon}
  ${news_list_item_title}
```

```
    ${news_list_item_summary}
    ${news_list_item_datetime}
  </message>

  <message id="vodd.user.news-item">
    ${news_item_picture}
    ${news_item_title}
    ${news_item_summary}
    ${news_item_text}
    ${news_item_datetime}
  </message>

</messages>
```

From:
https://wiki.skylark.tv/ - **wiki.skylark.tv**

Permanent link:
https://wiki.skylark.tv/manual/vod

Last update: **2020/01/14 08:38**