Matterhorn Engage Player

This page will be used to investigate integration with the Engage player.

How Engage works

Demo of Engage in action: http://demo.opencastproject.org/engage/ui/watch.html?id=1733c580-1fb8-44c7-a6d8-e93577463cfe

Engage's watch.js extracts information from URL parameters or JSON to display video, annotation, segmentation and other things: http://opencast.jira.com/wiki/display/MH/Engage+URL+Parameters

Engage expects the HTML page to have certain elements in order to work, as evident in watch.js and init-watch.js.

Looking at the demo, we see

- watch.js gets `/engage/ui/json/servicedata.json`, sets `segmentsUIURL = data.plugin_urls.segments_ui`, which is `/search/episode.json`
- segments_ui.js makes ajax call to `Opencast.Watch.getSegmentsUIURL()`, using data: `'id=' + mediaPackageId`
- from returned `data`, passes `data['search-results'].result` as `data` to segments_ui-plugin.js using `addAsPlugin()`,
- `segments_ui_dataMPMedia = data.mediapackage.media`, gets tracks info from `segments_ui_dataMPMedia.track` and
- write `track.url` to element `oc-vide-presenter-delivery-x-flv-rtmp` in watch.html
- watch.js then reads this element to set media URL using `Opencast.Player.setMediaURL`
- which calls `Videodisplay.setMediaURL`.
- End of clunky process.

Json examples. Theoretically, we could modify these REST endpoints to be pointing to Matterhorn server and the full featured Engage would just work, provided we don't run into cross-domain problems. JSONP helps with the crossdomain problems, with the exception of usertracking.

servicedata.json

```
{
 "plugin_urls":
 {
 "analytics": "/usertracking/footprint.xml",
 "annotation": "/annotation/annotations.json",
 "description":
 {
 "episode": "/search/episode.json",
 "stats": "/usertracking/stats.json"
 },
 "search": "/search/episode.json",
 "segments_text": "/search/episode.json",
 "segments_ui": "/search/episode.json",
 "segments": "/search/episode.json",
 "series":
 {
 "series": "/search/series.json",
 "episode": "/search/episode.json"
 }
 },
 "mediaDebugInfo":
 {
 "mediaPackageId": "",
 "mediaUrlOne": "",
 "mediaUrlTwo": "",
 "mediaResolutionOne": "",
 "mediaResolutionTwo": "",
 "mimetypeOne": "",
 "mimetypeTwo": ""
 }
}
```

Building the Engage player

http://opencast.jira.com/wiki/display/MH/Engage+Media+Player

```
svn checkout https://opencast.jira.com/svn/MH/trunk/
cd trunk/modules/matterhorn-engage-player/
mvn clean install
cd ../../shared-resources/player/
mv matterhorn-engage-player-1.4-SNAPSHOT.swf Videodisplay.swf
```

Additional operation in the pom.xml file can copy the player files structure to a new directory, suitable to use with Hydrant.

Testing the Engage player

```
cd ../../modules/matterhorn-engage-ui/
mvn install
```

Deploy the app?

Modify the servicedata.json to use test data

```
"mediaUrlOne": "http://demo.opencastproject.org/static/1733c580-1fb8-44c7-a6d8-e93577463cfe/2e3340ad-d309-4238-
b85b-eb09e06a2bc4/presenter.flv",
"mediaUrlTwo": "http://demo.opencastproject.org/static/1733c580-1fb8-44c7-a6d8-e93577463cfe/2149f073-abde-46ef-
a828-91c8a2016749/presentation.flv",
```

How to take out an Engage component

Figure out which HTML elements we don't need and have javascript calls to detach() them.

Engage structure

Videodisplay.swf

Actual video player written in ActionScript

FABridge.js

A proxy between JS and ActionScript functions

Videodisplay.js

`Videodisplay`

Contains basic events & functions to interact with the video player.

Events:

- VideodisplayReady()
- initialized()

Functions:

- play()
- stop()
- pause()
- mute()
- skipBackward()
- rewind()
- stopRewind()
- fastForward()
- stopFastForward()
- skipForward()
- passCharCode(int)
- seek(number)
- mute()
- setVolumeSlider(number)
- setVolumePlayer(number)
- closedCaptions()
- setMediaURL(argCoverOne, argCoverTwo, argStringOne, argStringTwo, argMimetypeOne, argMimetypeTwo, argPlayerstyle, slideLength)
- setCaptionsURL(string)
- videoSizeControl(sizeLeft, sizeRight)
- getViewState()
- setMediaResolution(argWidthMediaOne, argHeightMediaOne, argWidthMediaTwo, argHeightMediaTwo, argMultiMediaContainerLeft)

player-multi-hybrid-initialize.js

`Opencast.Initialize`

Calls

- Opencast.segments.getCurrentSlideId()
- Opencast.segments.getSegmentSeconds(
- Opencast.Watch.seekSegment(sec)
- Opencast.Description.doToggleDescription()
- Opencast.Analytics.doToggleAnalytics()
- Opencast.Series.doToggleSeriesDropdown()

Initializes the player, setting divId...

- bindVidSize
- doResize
- dropdownVideo_open
- dropdown_timer
- setPlayerReady
- onPlayerReady
- init
- setMediaResolution

player-multi-hybrid.js

`Opencast.Player`

Calls

- Opencast.engage.setLoadProgressPercent(value)
- Opencast.ariaSpinbutton.jumpToRange(newVolume)
- Opencast.Initialize.bindVidSize()
- Opencast.Initialize.doResize()
- Opencast.segments.getSecondsBeforeSlide()
- Opencast.Watch.seekSegment(sec)
- Videodisplay.videoSizeControl

Has

- getShowSections: getShowSections,
- getDuration: getDuration,
- getCaptionsBool: getCaptionsBool,
- getCurrentPosition: getCurrentPosition,
- getMediaPackageId: getMediaPackageId,
- getSessionId: getSessionId,
- getCurrentVideoSize: getCurrentVideoSize,
- getViewState: getViewState,
- getHtmlBool: getHtmlBool,
- getCurrentTime: getCurrentTime,

// setter

- setDragging: setDragging,
- setMediaURL: setMediaURL,
- setCaptionsURL: setCaptionsURL,
- setBrowserWidth: setBrowserWidth,
- setCCIconOn: setCCIconOn,
- setCCIconOff: setCCIconOff,
- setPlayerVolume: setPlayerVolume,
- setPlayPauseState: setPlayPauseState,
- setCurrentTime: setCurrentTime,
- setTotalTime: setTotalTime,
- setMediaPackageId: setMediaPackageId,
- setUserId: setUserId,
- setSessionId: setSessionId,
- setOptionClassName: setOptionClassName,
- setDuration: setDuration,
- setPlayhead: setPlayhead,
- setProgress: setProgress,
- setVolumeSlider: setVolumeSlider,
- setVideoSizeList: setVideoSizeList,
- setShareTimeLink: setShareTimeLink,

// do

- hideAll: hideAll,
- doToggleNotes: doToggleNotes,
- doToggleTranscript: doToggleTranscript,
- doToggleEmbed: doToggleEmbed,
- doToggleShareTime: doToggleShareTime,
- doToggleShare: doToggleShare,
- doToggleTimeLayer: doToggleTimeLayer,
- doToggleShortcuts: doToggleShortcuts,
- doToggleDownloads: doToggleDownloads,
- doTogglePlayPause: doTogglePlayPause,
- doToggleMute: doToggleMute,
- doPlay: doPlay,
- doPause: doPause,
- doFastForward: doFastForward,
- doSkipBackward: doSkipBackward,
- doRewind: doRewind,
- doSkipForward: doSkipForward,
- doSeek: doSeek,
- doToogleClosedCaptions: doToogleClosedCaptions,

// show

- showShare: showShare,
- showEditTime: showEditTime,
- showEmbed: showEmbed,
- showShareTime: showShareTime,

// hide

- hideShortcuts: hideShortcuts,
- hideShare: hideShare,

- hideEmbed: hideEmbed,
- hideShareTime: hideShareTime,

// video size control

- videoSizeControlSingleDisplay: videoSizeControlSingleDisplay,
- videoSizeControlAudioDisplay: videoSizeControlAudioDisplay,
- videoSizeControlMultiOnlyLeftDisplay: videoSizeControlMultiOnlyLeftDisplay,
- videoSizeControlMultiOnlyRightDisplay: videoSizeControlMultiOnlyRightDisplay,
- videoSizeControlMultiBigRightDisplay: videoSizeControlMultiBigRightDisplay,
- videoSizeControlMultiBigLeftDisplay: videoSizeControlMultiBigLeftDisplay,
- videoSizeControlMultiDisplay: videoSizeControlMultiDisplay,
- videoSizeControlSinglePlayerWithSlides: videoSizeControlSinglePlayerWithSlides,

// alert

- addAlert: addAlert,
- removeOldAlert: removeOldAlert,

// sound

- lowSound: lowSound,
- noneSound: noneSound,
- highSound: highSound,
- muteSound: muteSound,

// play/pause

- PlayPauseMouseOver: PlayPauseMouseOver,
- PlayPauseMouseOut: PlayPauseMouseOut,
- PlayPauseMouseDown: PlayPauseMouseDown,

// misc

- isPlaying: isPlaying,
- shareOverlayDisplayed: shareOverlayDisplayed,
- shortcutOverlayDisplayed: shortcutOverlayDisplayed,
- refreshScrubberPosition: refreshScrubberPosition,
- embedIFrame: embedIFrame,
- stopRewind: stopRewind,
- stopFastForward: stopFastForward,
- hideEditTime: hideEditTime,
- editTime: editTime,
- currentTime: currentTime,
- flashVars: flashVars,
- addEvent: addEvent,
- addFootprint: addFootprint

player-multi-hybrid-flash.js

`Opencast.FlashVersion`

Calls

- Opencast.engage.getPlayerType()

Has

- AC_Generateobj(objAttrs, params, embedAttrs): creates actual `<object>` to embed using `document.write(str)`
- AC_FL_RunContent(args)
- AC_GetArgs(args, ext, srcParamName, classid, mimeType)

engage-ui.js

`Opencast.engage`

Calls

- Opencast.Player.getDuration()

Has

- getCookie
- getPlayerType
- getLoadProgress
- setLoadProgressPercent
- getSearchServiceEpisodeIdURL
- getSearchServiceEpisodeJsonURL

watch.js

`Opencast.Watch`

Calls

- Opencast.Player.getDuration()

Has

- getAnalyticsURL
- getAnnotationURL
- getDescriptionEpisodeURL
- getDescriptionStatsURL
- getSearchURL
- getSegmentsTextURL
- getSegmentsUIURL
- getSegmentsURL
- getSeriesSeriesURL
- getSeriesEpisodeURL
- onPlayerReady
- seekSegment
- continueProcessing
- durationSet
- getClientShortcuts

Plugins

Analytics

Annotation

Bookmarks

Description

Download

EmbedControlHide

Logging

Search

Segments

Segments Text

Segments UI

Series

Embeded mode

Shows the player without Matterhorn Media Module components.

Replace 'watch.html' with 'embed.html' http://someURL:8080/engage/ui/watch.html?id=someMediaPackageId -> http://someURL:8080/engage/ui/embed.html?id=someMediaPackageId

Url parameters: http://opencast.jira.com/wiki/display/MH/Engage+URL+Parameters