
Ooyala Video Plugin

A Technical and User Guide

SEPTEMBER 25, 2012

Table of Content

DRAFT

OVERVIEW	3
INSTALL	3
RE-INSTALLING/UPDATING THE PACKAGE:	8
USING THE COMPONENT	9
USING THE CONTENT FINDER	10
UPLOADING A VIDEO	11
TECHNICAL OVERVIEW	14
OOYALA CQ5 INTEGRATION	14
MAVEN/ECLIPSE PROJECT STRUCTURE	16
LOG FILES	16

Overview

This document describes the technical implementation of the Ooyala video component and content finder tab on Adobe's CQ 5.5. This functionality is packaged following the "Package Manager" tool embedded in Adobe's CQ 5.5. The package contains the following items:

1. Content Finder Tab
2. Ooyala Video Component
3. Ooyala Configuration Page
4. OSGI Java Classes

The Ooyala plugin enables content authors to find videos from Backlot and place those on pages. Users can configure the video component with players and player sizes. The V2 API's are used to create the Ooyala plugin and are detailed at the following location:

<http://api.ooyala.com/docs/v2/>

INSTALL

The Ooyala CQ5 plugin (the plugin) has been developed using CQ5's standard component model. The plugin consists of an OSGi bundle/module to handle Ooyala Backlot/REST API integration, the Ooyala Video Component (jsp's, Dialogs, and supporting files), and the Tools Configuration Page. The plugin is distributed as a CQ5 Package for easy installation. The source code is also available for download from <https://github.com/ooyala/Ooyala-AdobeCQ/>

The Ooyala plugin, once downloaded is named "Ooyala-AdobeCQv01.zip". Do not unzip the file; Adobe CQ5.5 package manager expects it as a zip file.

Click on "package" on the right hand of the cq5.5 start up screen. To navigate directly to the CQ5 Package Manager Page the url is [http://\[replace with the author instance domain\]:4502/crx/packmgr/index.jsp](http://[replace with the author instance domain]:4502/crx/packmgr/index.jsp)

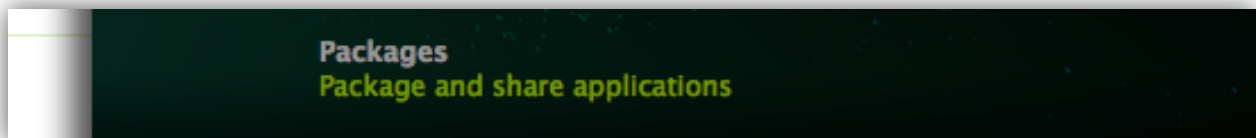


Figure 1

On the package manager screen click “Upload Package” on the top menu bar; this will bring up a wizard to upload the package. Navigate to the folder where the zip file is located. Ensure the zip file is not unzipped. When selected the name of the zip file will be displayed in the Package field. If this is the first time the package is being uploaded the “Force Upload” can be left blank. In general it is recommended to click this button to install new versions of the package. Uninstalling the package will delete all the data as well.

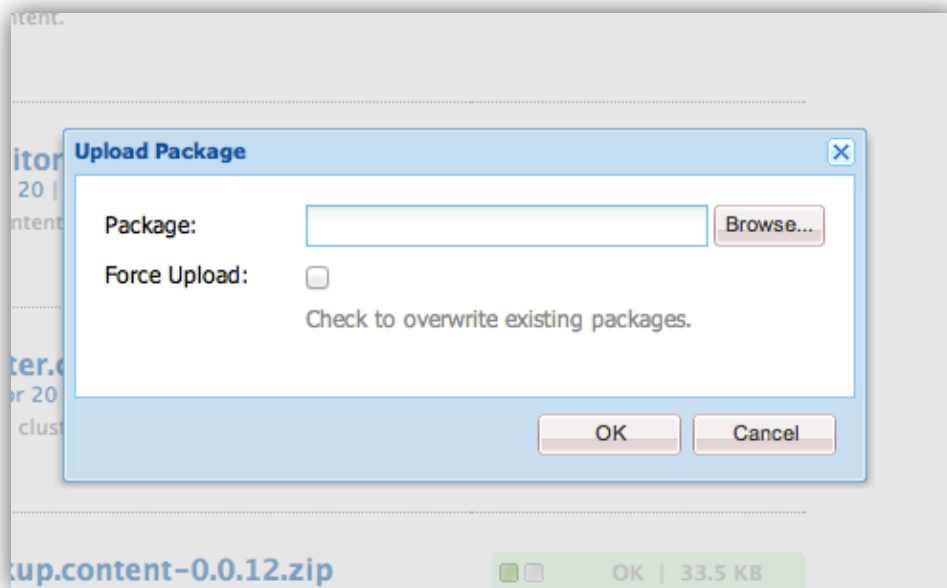


Figure 2

Once the zip file has been uploaded, it needs to be installed. Click on the “install” button on the top right.

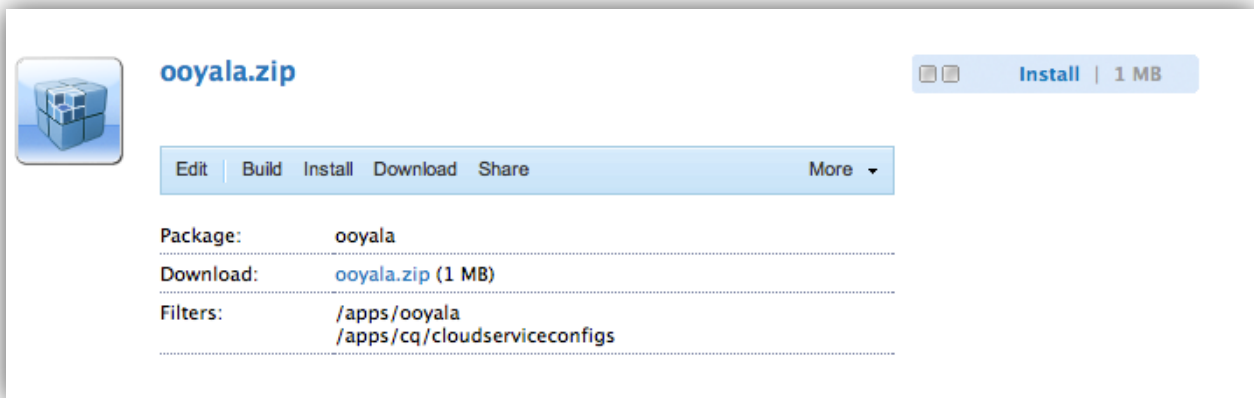


Figure 3

The wizard below will can be left to defaults. Click on the “Install” button.

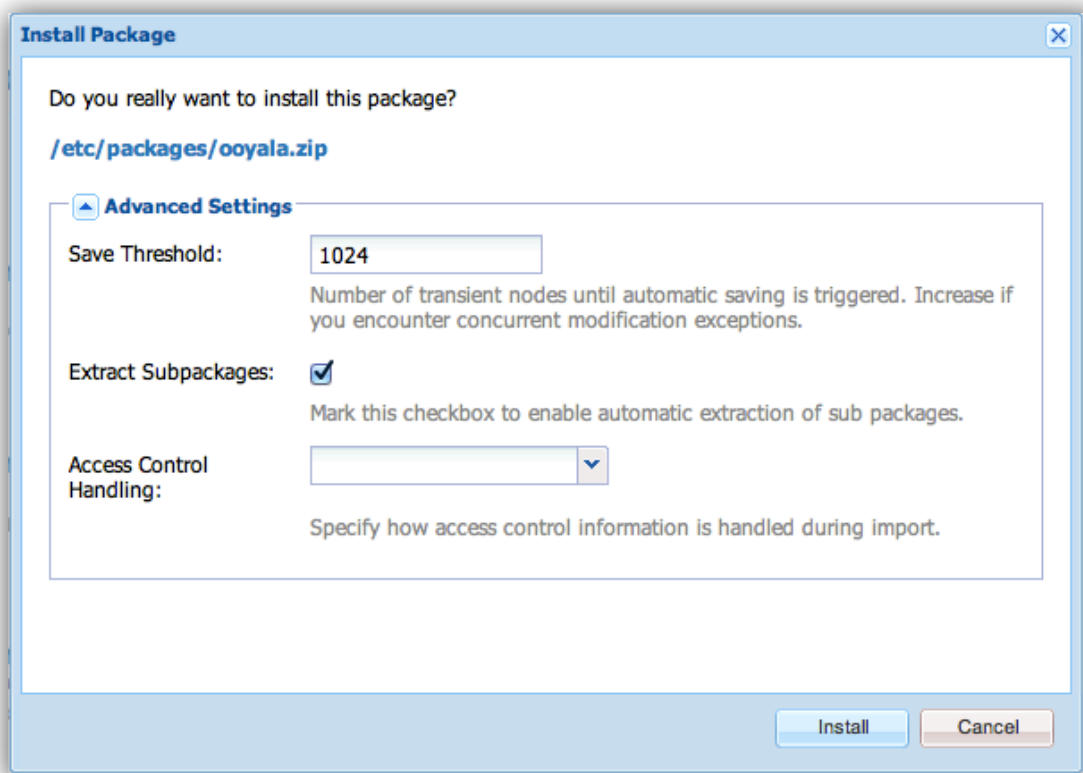


Figure 4

If the install was successful the following message will be displayed at the bottom of the console.

```
A /apps/cq/cloudserviceconfigs/components/configpage/  
A /apps/cq/cloudserviceconfigs/components/configpage/  
A /apps/cq/cloudserviceconfigs/components/configpage/  
A /etc/cloudservices/ooyala  
A /etc/cloudservices/ooyala/jcr:content  
saving approx 90 nodes...  
Package imported.  
  
Package installed in 1960ms.
```

Figure 5

Once the package installs successfully, proceed back to the main screen and click on “Tools”. On the tools screen navigate to the “Cloud Services Configurations” folder. There should be a page titled “Ooyala Configuration”.

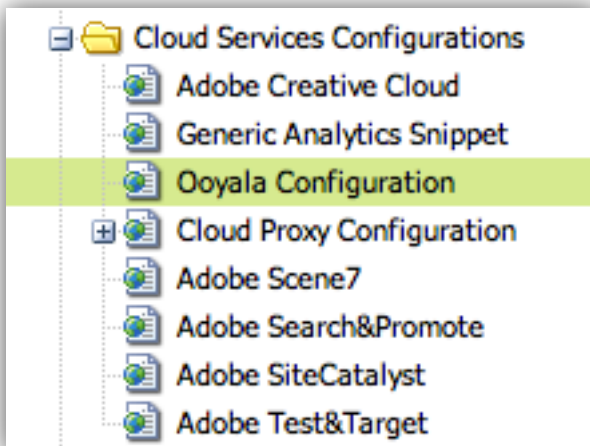


Figure 6

Double click on this page will open up the Ooyala configuration dialog. The dialog should load automatically, or the user can click the “Edit” button.

Ooyala Configuration

Configuration | Edit

Figure 7

The following fields are displayed when the dialog loads:

- API Key: the global account key from backlot
- API Secret : the global account key from backlot
- Metadata Keys: The Ooyala api matches metadata keys exactly, so a list of the actual key needs to be configured. This list of keys will be displayed on the content finder tab, so users can select which metadata field they would like to search on.

Edit Component

Ooyala Setup

Root Path:

API Key:

API Secret:

Metadata Keys: [Add Item](#)

Add metadata keys that should be searchable from the content finder.

OK Cancel

Figure 8

RE-INSTALLING/UPDATING THE PACKAGE:

Re-installation of the Ooyala package requires Uninstalling the Ooyala OSGi bundle first. Once the OSGi bundle has been uninstalled, the package can be re-installed using the same instructions as the previous (Installation) section.

WARNING: Re-installing the Ooyala CQ5 Package will erase the Ooyala Backlot API Credentials on the CQ5 Tools Page. Make sure to backup your API Key/s, API Secret, and Metadata fields before re-installing the package.

This procedure requires CQ5 administrator privileges.

Uninstall the Ooyala OSGi bundle:

- Navigate to the CQ5 OSGi Web Console: <http://localhost:4502/system/console/bundles>
- Using your browser's search function, find the Ooyala OSGi bundle. Search for "ooyala".
- The bundle should be named ooyala-osgi com.siteworx.cq5.ooyala.osgi
- Making sure you have selected the Ooyala OSGi bundle.
- Click on the trash can icon/uninstall.

WARNING: Deleting an incorrect bundle could cause the system to fail.

Re-install the Ooyala CQ5 Package using the instruction from the previous section.

USING THE COMPONENT

To use the Ooyala video component on any page, the page container component needs to allow the “Ooyala Video Component”. This is done by clicking on the Design button located at the bottom of the Sidekick.

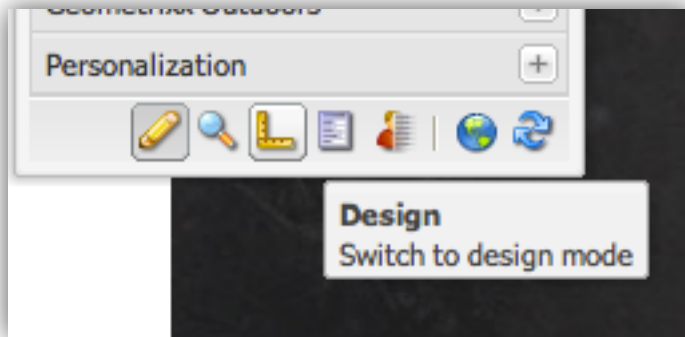


Figure 9

Check the “Ooyala Video Component” under the Allowed Components “General” Category

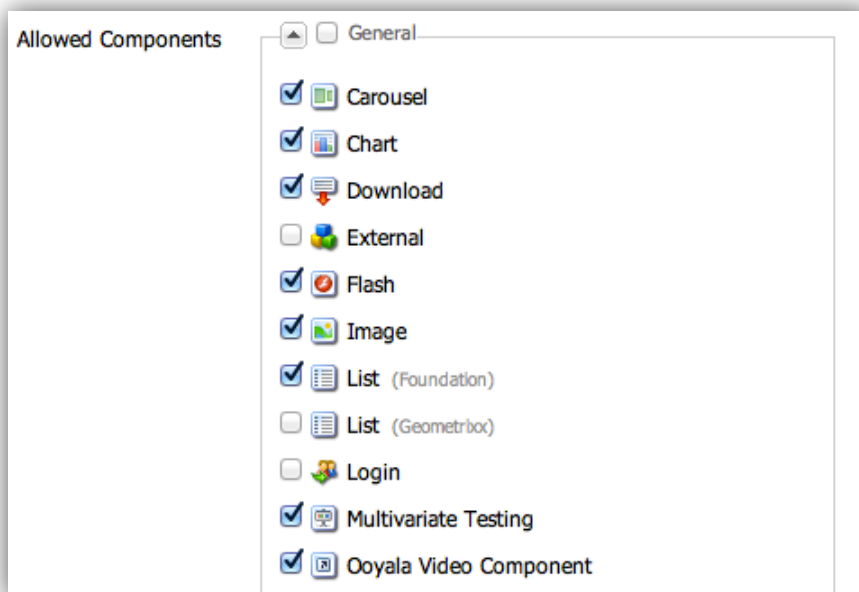


Figure 10

If everything was successful, back in edit mode, the sidekick should now display the “Ooyala Video Component” for drag and drop on pages.

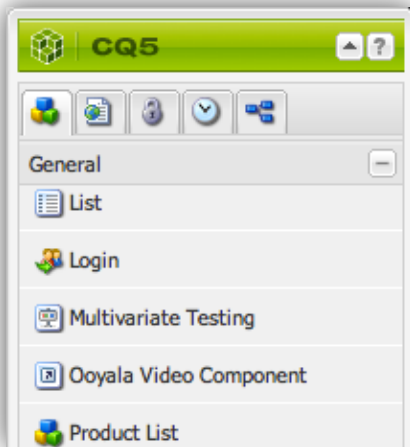


Figure 11

When the Ooyala video component is dropped on a page, the component renders the following message. The next step is to find a video using the content finder and drag and drop that video on top of this component. When the video is dropped the Video ID will be automatically populated. The next step is to double click on the component to bring up the player configuration. There will be a dropdown where users can choose the player and sizes of the video player.

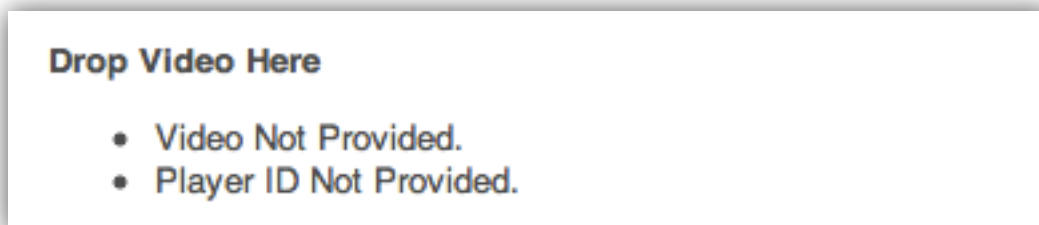


Figure 12

USING THE CONTENT FINDER

The content finder tab can be used for searching and filtering through Ooyala videos for easy placement on CQ pages. The content finder tab is available from any page in the authoring environment. In order to use the Ooyala content finder functionality, click on the tab at the top of the content finder. The Ooyala tab will be displayed as shown.

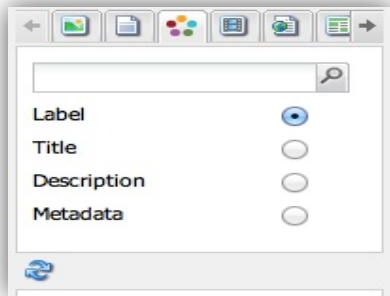


Figure 13

The content finder offers four different search functionalities; search by label, by title, by description, and by metadata. Search by title and by label accept plain text queries and return the results that best match. Search by label allows the user to enter a label used on their Ooyala account and returns all those videos tagged with that label. The search by label requires an exact spelling match in order to function; for this reason, it provides the user with suggestions in an auto-complete box as shown.

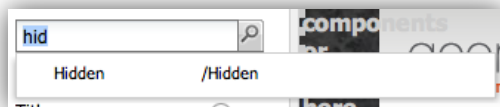


Figure 14

Searching by metadata shows suggestions in the same manner, based on metadata keys entered by an administrator on the Ooyala configuration screen. Searching by metadata requires both a key and a value. Choosing a key from the autocomplete prefills the search field with the key and a colon(:). Users should enter the value they wish to search for within that key after the colon, before submitting their query. Remember, it is important to make use of the autocomplete functionality to ensure your search is using the correct spelling and returning the proper results.

UPLOADING A VIDEO

Videos can be uploaded directly from CQ. Under the Tools section there is a folder labeled “Ooyala”. Double click on the “Ooyala Uploader” page will launch the uploader widget.



Figure 15

On the page there are three call to actions, the name and description are not required:

1. Name (text field)
2. Description (text field)
3. Select Asset (button)

The select asset button will launch a native file uploader, where the user could navigate their file system and find the video to upload.

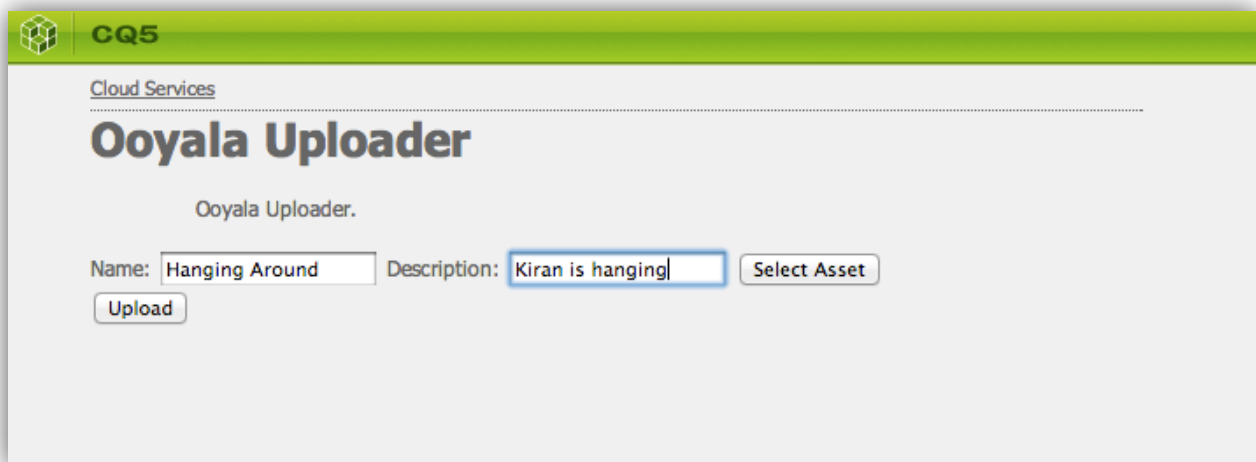
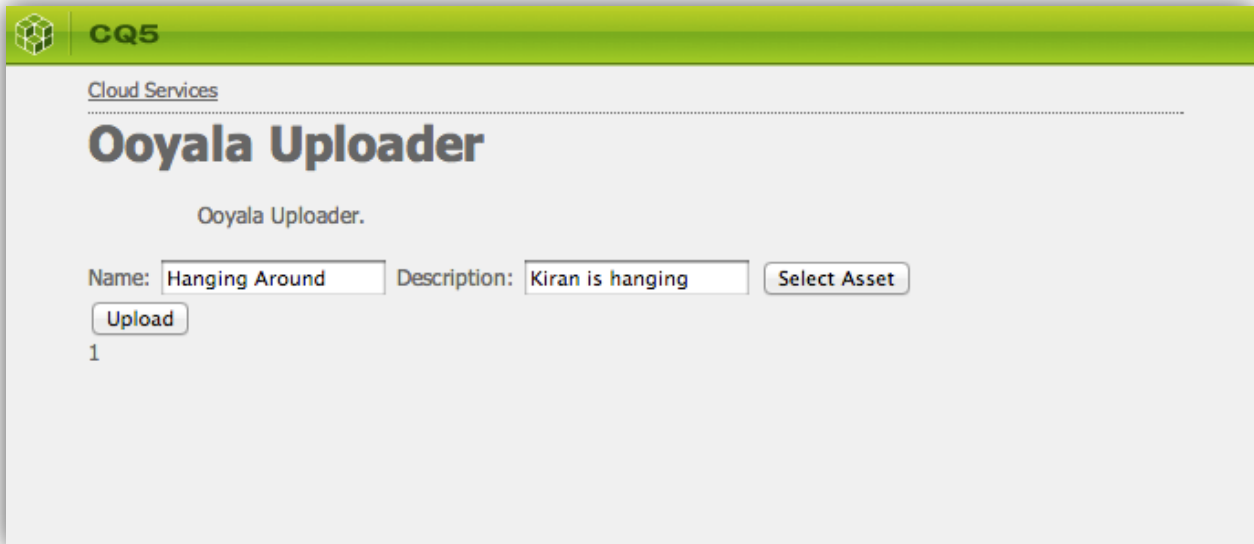


Figure 16

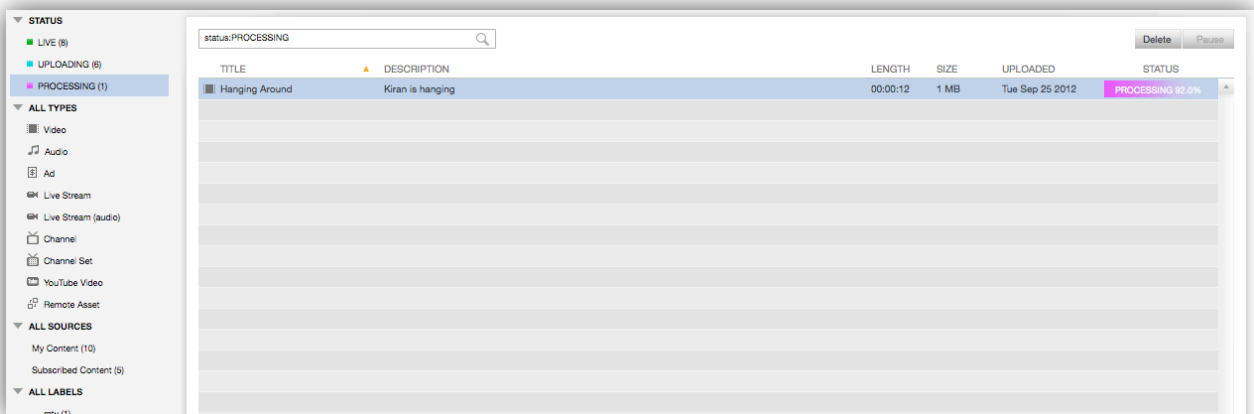
Once the video has been selected and the proper name and description have been added, click the Upload button. The upload button will take the video, and create chunks that will then be uploaded into Ooyala's and in a few minutes users could find the video in Backlot. A numeric value starting with 0, 0.5, 1 will be displayed.



The screenshot shows the CQ5 Ooyala Uploader interface. At the top, there is a green header with the CQ5 logo. Below the header, the text "Cloud Services" is visible. The main heading is "Ooyala Uploader". Underneath, it says "Ooyala Uploader." There are two input fields: "Name:" with the value "Hanging Around" and "Description:" with the value "Kiran is hanging". To the right of the description field is a "Select Asset" button. Below the name field is an "Upload" button. At the bottom left, the number "1" is displayed.

Figure 17

When the number 1 is displayed the video has been uploaded and Backlot is processing the file.



The screenshot shows the CQ5 asset management interface. On the left, there is a sidebar with navigation options: STATUS (LIVE (8), UPLOADING (6), PROCESSING (1)), ALL TYPES (Video, Audio, Ad, Live Stream, Live Stream (audio), Channel, Channel Set, YouTube Video, Remote Asset), ALL SOURCES (My Content (10), Subscribed Content (6)), and ALL LABELS (mtv (1)). The main area displays a table of assets. The first asset is "Hanging Around" with the description "Kiran is hanging". It has a length of 00:00:12, a size of 1 MB, and was uploaded on Tue Sep 25 2012. The status is "PROCESSING 92.0%". There are "Delete" and "Pause" buttons in the top right corner of the table.

TITLE	DESCRIPTION	LENGTH	SIZE	UPLOADED	STATUS
Hanging Around	Kiran is hanging	00:00:12	1 MB	Tue Sep 25 2012	PROCESSING 92.0%

Figure 18

TECHNICAL OVERVIEW

The following is a class diagram for the Ooyala video component and content finder functionality. These classes

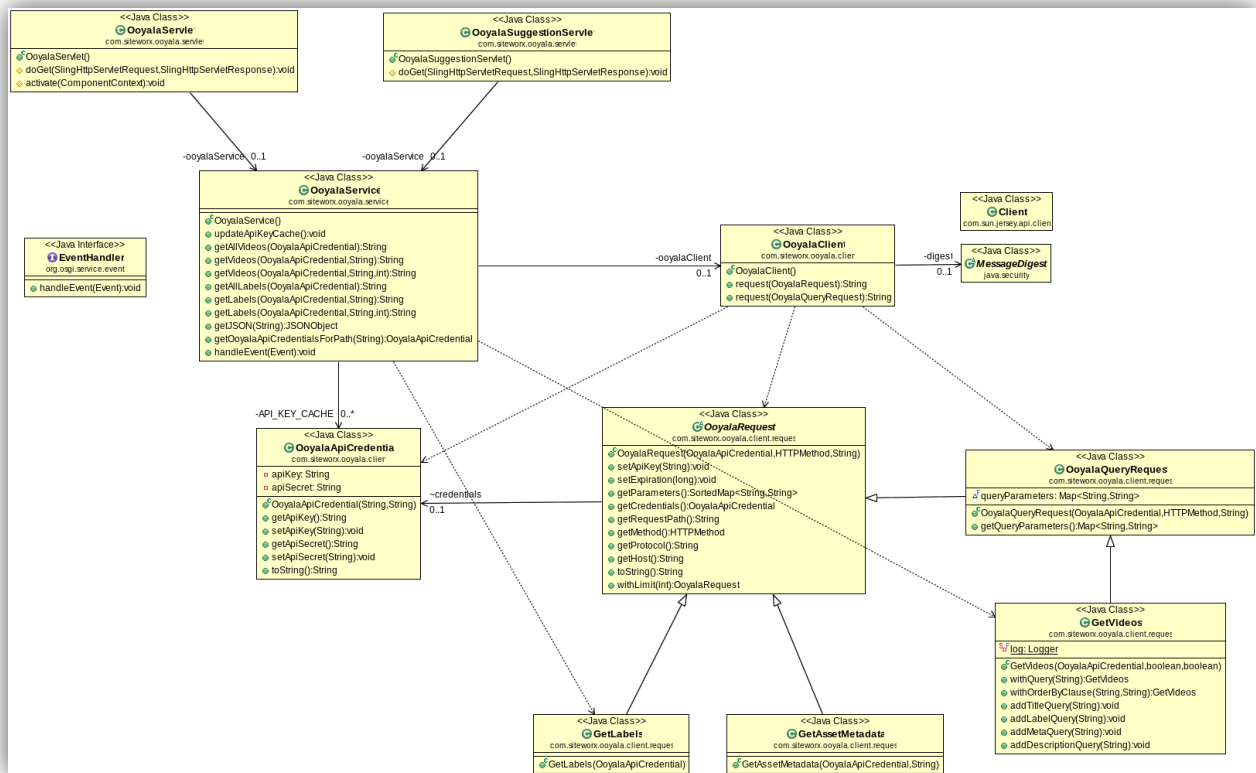


Figure 19

Ooyala CQ5 Integration

The following classes were developed as part of the solution

Class	Description
OoyalaApiCredential.java	Contains one API key/secret pair.
OoyalaService.java	OSGi Service. Provides methods to call Ooyala API (REST). Has methods such as getVideos, getLabels,

Class	Description
	<p>etc.</p> <p>Methods take a OoyalaApiCredential, this allows for multiple credentials (per site/path)</p>
OoyalaClient.java	Handles low level implementation of REST calls. Uses Jersey client internally.
OoyalaRequest.java OoyalaQueryRequest.java	<p>These (abstract) classes encapsulate base parameters for a Ooyala REST API request.</p> <p>Subclasses add more specific parameters/behavior like limit, order by, etc.</p>
OoyalaServlet.java	<p>Handles search/browse requests from the CQ Content Finder. The Delegates to OoyalaService.</p> <ul style="list-style-type: none"> responds to: GET /bin/wcm/ooyala?query=[query]&limit=[limit] returns a CQ-ContentFinder-formatted JSON response.
OoyalaSuggestionServlet.java	<p>Handles autocomplete/suggestion requests from the CQ Content Finder. Delegates to OoyalaService.</p> <ul style="list-style-type: none"> reponds to: GET /bin/wcm/ooyala/suggestions?query=[query] returns a CQ-ContentFinder-formatted JSON response.

The following components are included in the plugin package:

Component	Description
ooyalaVideo (apps/ooyala/components/content/ooyalaVideo)	A Standard methodology to creating CQ5 components. When used on a page, It is placed on a cq:Page wherever a Video should be displayed.

Component	Description
ooyalaTab.js (apps/ooyala/wcm/extensions/contentFinder/ooyalatab.js)	This is the custom Tab for the Content Finder. It contains JavaScript configuration for the Tab. This is an instance of the class in cq "CQ.wcm.ContentFinderTab, information about this class can be found in at "http://dev.day.com/docs/en/cq/current/widgets-api/index.html"
etc/cloudservices/ooyala	This is a configuration page/node exposed through the Tools section in CQ. It is used to configure the API key(s)/secret(s) in OoyalaService.java.

Maven/Eclipse Project Structure

Structure	Description
CQPACKAGE	The CQ Package available for installation
DOCUMENTATION	Documentation
OSGI	The back end services
UI	The code for the component and front end.

To compile the code, a mvn clean install in the parent directory of where the code is checked out. Then navigate to the CQPackage and run the build CQ package. This will create a zip file read to upload into Adobe CQ 5.5.

Log files

CQ reports its current state into its log files, and it is often the best way to debug errors in the

system:

File Name	File Location	Description
access.log	\$CQ_HOME/logs	Contains reports of all incoming HTTP requests
error.log	\$CQ_HOME/logs	Contains standard log information along with any error messages that had occurred.
request.log	\$CQ_HOME/logs	Contains all incoming requests/responses.
stdout.log	\$CQ_HOME/logs	Contains messages created with System.out.println(...).
stderr.log	\$CQ_HOME/logs	Contains messages created with System.err.println(...).
server.log	\$CQ_HOME/logs	Contains messages from other parts of CQ, like the OSGi bundles and starting up the instance.
error.log	\$CQ_HOME/logs/crx	Contains errors in \$CQ_HOME/logs/crx/error.log
access.log	\$CQ_HOME/server/logs	Similar to access.log in \$CQ_HOME/logs
cq.pid	\$CQ_HOME/server/logs	Contains the PID of the running CQ instance.
startup.log	\$CQ_HOME/server/logs	Contains messages that CQ logged during startup.