

Streaming (Pseudo) Video In Grails



Ryan Vanderwerf
Chief Architect
ReachForce
www.reachforce.com

Streaming Video In Grails

My Background

- Currently building a Grails and Cloud based infrastructure for ReachForce
- Architected a Grails solution for Developerprogram.com that allows rapid deployment of Developer Program portals for all kinds of companies, specializing in the mobile industry.
- Built Java and Linux based webcasting for events such as SXSW, built telecom software, and ASP's for the financial sector
- Worked with Java since 1996, and built server-side applications ever since
- Enticed into the Groovy and Grails space by speakers at the early NFJS conferences

Streaming Video In Grails

What we will cover

- Pseudo Streaming Video Plugin
- Plugin layout
- Video upload process
- Video transformation (Basic ffmpeg)
- Flowplayer and Jw-Flv examples
- Taglib usage
- Distributed video processing via Quartz

Streaming Video In Grails

➤ Common need – light video streaming for Grails site

➤ **Options**

- Youtube
- Vimeo
- Flickr
- Slideshare
- Screencast
- Animoto
- ... and more

Yes they are great but.....

Streaming Video In Grails

- What about security?
- **What if you want your users to be able to upload video assets into your app?**
- **If they do upload, how are you going to manage all kinds of video formats and stream them?**
- **How can I handling streaming if users upload all kinds of different formats?**
- **How do I divide up the load as to not slow down my server after they upload them?**

Streaming Video In Grails

- What about security?
- **What if you want your users to be able to upload video assets into your app?**
- **If they do upload, how are you going to manage all kinds of video formats and stream them?**
- **How can I handling streaming if users upload all kinds of different formats?**
- **How do I divide up the load as to not slow down my server after they upload them?**

Streaming Video In Grails

Video Plugin History –

Abandonware created by Cantina Consulting
made for Grails 1.0 RC1 - 2007

Fixed up for Grails 1.3.X by KPI Consulting
(Ryan Vanderwerf and Shawn McKisson) – 2011

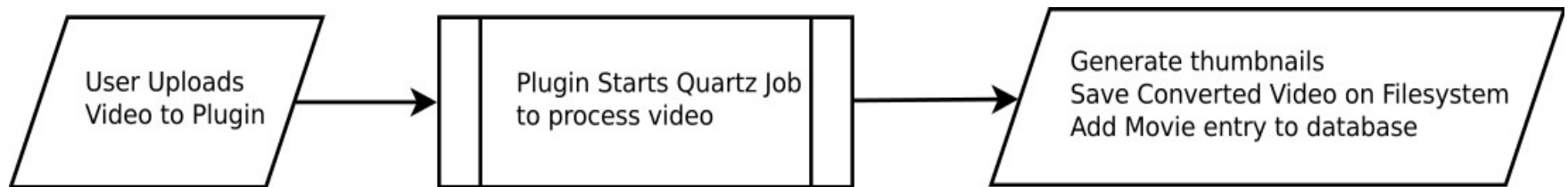
– Also fixed streaming, used Tomcat byte range
support so mp4 streaming will work

Fixed up for Grails 2.0.x by Ryan – 2012

Updated to use JWFLV 5.10

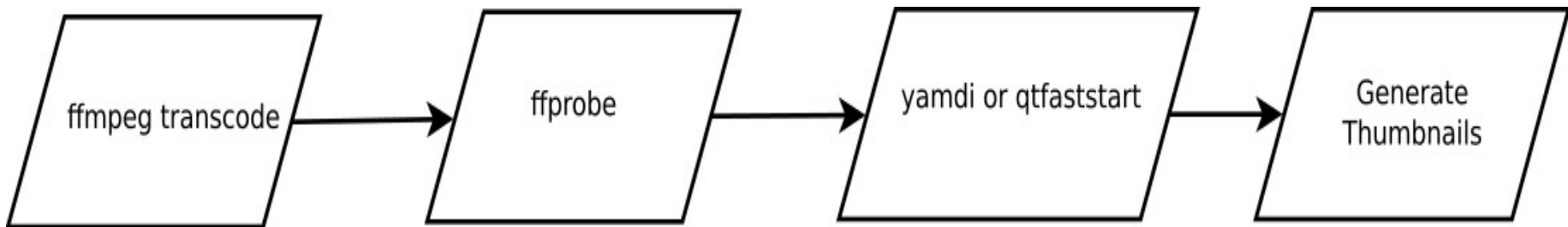
Streaming Video In Grails

Video Upload Process



Streaming Video In Grails

Video Transformation Process



Streaming Video In Grails

Answers to our questions.. yes I want to add video support into my application.

- Security: Override either of these actions In MovieController
 - streamflv (Streaming FLV only)
 - display (mp4)
- Add your security to your own actions with either Spring Security @Secured closure
- Use SpringSecurity or other framework and protect via URI like
- Feel free to take what you need out of those methods or extend them to do any specialty work

Streaming Video In Grails

FFMpeg (bundles with libavcodec) – quick overview

- Parameters may vary based on version you are using
- ex. `conversionArgs = "-b 600k -r 24 -ar 22050 -ab 96k"`
 - “-b 600k” set video bitrate of video to 600kbps
 - “-r 24” set framerate to 24 fps
 - “-ar 22050” audio sampling frequency rate
 - “-ab 96k” audio bit rate

Streaming Video In Grails

Managing Video formats

- Shells and runs ffmpeg to transcode video into common format
- Users can upload anything you can throw at ffmpeg, the most common video transcoding engine available (You can plug something else in without too much trouble)
- Uses YAMDI to extract metadata from video
- Uses ffprobe to extract mp4 data from footer and move to front of file to make streaming work
- Has support out of the box for Flowplayer and Jw-Flv

Streaming Video In Grails

Distribute the load

- Uses Quartz plugin to transcode uploads
- Pair with Clustered Quartz or Terracotta to distribute the load for large number of uploads (or offload to a separate set of servers that just process video)
- See my distributed Quartz slides for setting up JDBCJobStore with clustering or Terracotta setup

Streaming Video In Grails

JW-FLV Player

<http://www.longtailvideo.com/players/jw-flv-player/>

Open sourced flash and HTML5 video player, Cross platform playback

2 ways to embed:

```
<!-- START OF THE PLAYER EMBEDDING TO COPY-PASTE -->
<div id="mediaplayer">JW Player goes here</div>

<script type="text/javascript" src="jwplayer.js"></script>
<script type="text/javascript">
    jwplayer("mediaplayer").setup({
        flashplayer: "player.swf",
        file: "video.mp4",
        image: "preview.jpg"
    });
</script>
<!-- END OF THE PLAYER EMBEDDING -->
```

Streaming Video In Grails

JW-FLV Player

<http://www.longtailvideo.com/players/jw-flv-player/>

Or use taglib of plugin:

```
<vid:display movie='${movie}' player="jwflv" stream='true'/>
```

Parameters:

movie	Movie domain object
id	Id of movie object
stream	true/false stream or download
player	jwflv or flowplayer

Streaming Video In Grails

Flowplayer

<http://flowplayer.org/demos/plugins/streaming/index.html>

Or Use taglib of plugin: (currently uses older version of flowplayer)

```
<vid:display movie='${movie}' player="flowplayer" stream='true'/>
```

Parameters:

movie	Movie domain object
id	Id of movie object
stream	true/false stream or download
player	jwflv or flowplayer

Streaming Video In Grails

```
video{
    location="/tmp/" // or shared filesystem drive for a cluster
    yamdi{
        path="/usr/bin/yamdi" // FLV metadata injector (IF TYPE= FLV)
    }
    ffmpeg {
        fileExtension = "flv" // use flv or mp4
        conversionArgs = "-b 600k -r 24 -ar 22050 -ab 96k"
        path="/usr/bin/ffmpeg"
        makethumb = "-an -ss 00:00:03 -an -r 2 -vframes 1 -y -f mjpeg"
    }
    ffprobe {
        path="/usr/bin/ffprobe" // finds length of movie
        params=""
    }
    flowplayer {
        version = "3.1.2" // use empty string for no version on file
    }
    swfobject {
        version = "" // used for jw-flv player
    }
    qtfaststart {
        path= "/usr/sbin/qtfaststart" // if type == mp4 rearrange metadata
    }
}
```

Streaming Video In Grails

Demo

The screenshot shows a web browser window with the address bar at `localhost:8080/video/movie/show/1`. The browser's tab is titled "Show Movie". The page has a navigation bar with links for "Home", "Movie List", and "New Movie". Below the navigation bar, the page title "Show Movie" is displayed. A message box at the top of the content area states "Movie 1 created.".

The main content area displays the following movie details:

- Id:** 1
- Create Date:** 2012-07-16 00:37:48.856
- Created By:**
- Description:** fdasdafs
- File Name:** big_buck_bunny.mp4
- Content Type:** {}
- Play Time:** 1 minute
- Size:** 5.21 Mb
- Video:**

Below the details, there are four video player placeholders arranged in a 2x2 grid:

- Http streaming (jwflv):** A video player showing a landscape with pink clouds and green trees. The play button is visible in the center. The progress bar at the bottom shows 00:00.
- Full file download (jwflv):** A video player showing the same landscape as the first player. The play button is visible in the center. The progress bar at the bottom shows 00:00.
- Http streaming (flowplayer):** A black video player placeholder.
- Full file download (flowplayer):** A black video player placeholder.

Streaming Video In Grails

Flowplayer Plugins

<http://flowplayer.org/documentation/api/index.html>

Flowplayer support various plugins to give special effects, enhanced control. They checked into google code project called 'flowplayer-plugins' and are broken up between flash and javascript plugins.

More info here:

<http://code.google.com/p/flowplayer-plugins/>

Streaming Video In Grails

More Information

<https://github.com/rvanderwerf/grails-video>

<http://terracotta.org/downloads/open-source/catalog>

<http://flowplayer.org/>

<http://www.longtailvideo.com/players/jw-flv-player/>

<http://ffmpeg.org/>

<http://code.google.com/p/flowplayer-plugins/>

Streaming Video In Grails

Contact Me

Via twitter: <https://twitter.com/RyanVanderwerf>

Google+/email: rvanderwerf@gmail.com

Blog: <http://rvanderwerf.blogspot.com>