









From: Ken Fallon <ken.fallon@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: Re: [Dev & Muffy] Atom feed question(s)  
Date: Mon, 15 Oct 2012 20:03:23 +0200  
X-Mailer: Claws Mail 3.8.1 (GTK+ 2.24.11; x86\_64-redhat-linux-gnu)

---

On Sun, 14 Oct 2012 09:46:01 -0400  
epicanis+hpr@dogphilosophy.net wrote:

> I've got a couple of bits of coding that I'm working on. One of them  
> For the latter, I'm assuming what's needed is a web-based form that  
> guides a submitter into filling in the appropriate metadata, summary,  
> etc.,

Yes that's correct.

> and then converts that into a valid individual item to be  
> incorporated into the Atom feed, is that correct?

It should add it to a /Entry element, that is top-level element of a  
stand-alone Atom Entry Document.

<https://tools.ietf.org/html/rfc4287#section-4.1.2>

4.1.2. The "atom:entry" Element

The "atom:entry" element represents an individual entry, acting as a  
container for metadata and data associated with the entry. This  
element can appear as a child of the atom:feed element, or it can  
appear as the document (i.e., top-level) element of a stand-alone  
Atom Entry Document.

> Or is it preferable

> that the Atom data be populated from elsewhere (reading tags from the  
> audio or from the .txt file or whatever)?

The form should be the source of truth and will override all other data  
with the possible exception of "enclosure" and "type".

<https://tools.ietf.org/html/rfc4287#section-4.2.7>

4.2.7.2.4

The value "enclosure" signifies that the IRI in the value of the href  
attribute identifies a related resource that is potentially large in  
size and might require special handling. For atom:link elements with  
rel="enclosure", the length attribute SHOULD be provided.

4.2.7.3. The "type" Attribute

On the link element, the "type" attribute's value is an advisory media  
type: it is a hint about the type of the representation that is  
expected to be returned when the value of the href attribute is  
dereferenced. Note that the type attribute does not override the  
actual media type returned with the representation. Link elements MAY  
have a type attribute, whose value MUST conform to the syntax of a MIME  
media type [MIMereg].

<https://tools.ietf.org/html/rfc4288>

<https://tools.ietf.org/html/bcp13>

> Is the current (rss) feed generated automatically from somewhere or  
> updated by hand?

It is generated by a php script developed by Droops and modified by  
myself.

> (The former is a more ambitious project to build a web-based system  
> for generating appropriately-formatted output audio from an uploaded  
> original.

> The initial implementation will start with taking a .wav file and  
> optional "album art" and input from a form for metadata

We may put links to guide lines where the preference would be flac, wav, ogg etc with bitrates and encoding, but the form should also upload any audio format at all. Even files that it cannot determine the mime type of. The logic is that it is better to have a low fidelity show than no show.

Album art should not be promoted. The reason is that we reserve fields "Album Art", "Album = Hacker Public Radio", and the first part of "Comment = <http://hackerpublicradio.org>" HPR branding on the version that we distribute. We will probably remove any embedded art as it is know to cause issues with media players. That should not be the case for the versions we upload to Archive.Org. All tagging in the media files will be added from the data submitted in the upload form.

The show notes can be in text, html or xhtml format and the encoding needs to be correct. The show notes may have images supplied so we need to be able to support multiple images.

Remember that nothing should be trusted in the upload. We have "hacker" in our name after all. The entire delivery package (media, xml file, image(s) ) should be placed in a non browsable directory, in a subfolder composed of  
<priority> - the priority of the show  
\_ - delimiter underscore  
<date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
\_ - delimiter underscore  
<uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool

This folder should not be disclosed to the uploader to prevent people using HPR as a file sharing backup solution.

For example:

1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af

You should also check the upload for javascript and all other variations of it encoded in it. There is a great book on php security that I recommend you read if you already haven't done so.

<http://www.phparch.com/books/phparchitects-guide-to-php-security/>

>, and using

> that to generate Opus audio, but once that's working I'll be adding  
> vorbis and possibly mp3 output along with input from most non-.wav  
> formats for the original. It sounds like that would be useful to  
> adapt for HPR use assuming I don't suck and can get it all working  
> usefully.)

The upload form and the post processing will be handled by different processes. This is for security, ease of maintenance and as it's the unix way.

The backend process will ssh into the frontend server and pull in the directory and scan it.

The processing of audio will be "just in time" so that we can add daily news items.

Ken.

>

> [Dev mailing list](#)

> [Dev@hackerpublicradio.org](mailto:Dev@hackerpublicradio.org)

> [http://hackerpublicradio.org/mailman/listinfo/dev\\_hackerpublicradio.org](http://hackerpublicradio.org/mailman/listinfo/dev_hackerpublicradio.org)



From: Ken Fallon <ken.fallon@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: Re: [Dev] [Dev & Muffy] Atom feed question(s)  
Date: Mon, 15 Oct 2012 20:03:23 +0200  
Sender: dev-bounces@hackerpublicradio.org  
X-Mailer: Claws Mail 3.8.1 (GTK+ 2.24.11; x86\_64-redhat-linux-gnu)

---

On Sun, 14 Oct 2012 09:46:01 -0400  
epicanis+hpr@dogphilosophy.net wrote:

> I've got a couple of bits of coding that I'm working on. One of them  
> For the latter, I'm assuming what's needed is a web-based form that  
> guides a submitter into filling in the appropriate metadata, summary,  
> etc.,

Yes that's correct.

> and then converts that into a valid individual item to be  
> incorporated into the Atom feed, is that correct?

It should add it to a /Entry element, that is top-level element of a  
stand-alone Atom Entry Document.

<https://tools.ietf.org/html/rfc4287#section-4.1.2>

4.1.2. The "atom:entry" Element

The "atom:entry" element represents an individual entry, acting as a  
container for metadata and data associated with the entry. This  
element can appear as a child of the atom:feed element, or it can  
appear as the document (i.e., top-level) element of a stand-alone  
Atom Entry Document.

> Or is it preferable

> that the Atom data be populated from elsewhere (reading tags from the  
> audio or from the .txt file or whatever)?

The form should be the source of truth and will override all other data  
with the possible exception of "enclosure" and "type".

<https://tools.ietf.org/html/rfc4287#section-4.2.7>

4.2.7.2.4

The value "enclosure" signifies that the IRI in the value of the href  
attribute identifies a related resource that is potentially large in  
size and might require special handling. For atom:link elements with  
rel="enclosure", the length attribute SHOULD be provided.

4.2.7.3. The "type" Attribute

On the link element, the "type" attribute's value is an advisory media  
type: it is a hint about the type of the representation that is  
expected to be returned when the value of the href attribute is  
dereferenced. Note that the type attribute does not override the  
actual media type returned with the representation. Link elements MAY  
have a type attribute, whose value MUST conform to the syntax of a MIME  
media type [MIMereg].

<https://tools.ietf.org/html/rfc4288>

<https://tools.ietf.org/html/bcp13>

> Is the current (rss) feed generated automatically from somewhere or  
> updated by hand?

It is generated by a php script developed by Droops and modified by  
myself.

> (The former is a more ambitious project to build a web-based system  
> for generating appropriately-formatted output audio from an uploaded  
> original.

> The initial implementation will start with taking a .wav file and

> optional "album art" and input from a form for metadata

We may put links to guide lines where the preference would be flac, wav, ogg etc with bitrates and encoding, but the form should also upload any audio format at all. Even files that it cannot determine the mime type of. The logic is that it is better to have a low fidelity show than no show.

Album art should not be promoted. The reason is that we reserve fields "Album Art", "Album = Hacker Public Radio", and the first part of "Comment = <http://hackerpublicradio.org>" HPR branding on the version that we distribute. We will probably remove any embedded art as it is know to cause issues with media players. That should not be the case for the versions we upload to Archive.Org. All tagging in the media files will be added from the data submitted in the upload form.

The show notes can be in text, html or xhtml format and the encoding needs to be correct. The show notes may have images supplied so we need to be able to support multiple images.

Remember that nothing should be trusted in the upload. We have "hacker" in our name after all. The entire delivery package (media, xml file, image(s) ) should be placed in a non browsable directory, in a subfolder composed of  
<priority> - the priority of the show  
\_ - delimiter underscore  
<date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
\_ - delimiter underscore  
<uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool

This folder should not be disclosed to the uploader to prevent people using HPR as a file sharing backup solution.

For example:

1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af

You should also check the upload for javascript and all other variations of it encoded in it. There is a great book on php security that I recommend you read if you already haven't done so.

<http://www.phparch.com/books/phparchitects-guide-to-php-security/>

>, and using

> that to generate Opus audio, but once that's working I'll be adding  
> vorbis and possibly mp3 output along with input from most non-.wav  
> formats for the original. It sounds like that would be useful to  
> adapt for HPR use assuming I don't suck and can get it all working  
> usefully.)

The upload form and the post processing will be handled by different processes. This is for security, ease of maintenance and as it's the unix way.

The backend process will ssh into the frontend server and pull in the directory and scan it.

The processing of audio will be "just in time" so that we can add daily news items.

Ken.

>

> [Dev mailing list](#)

> [Dev@hackerpublicradio.org](mailto:Dev@hackerpublicradio.org)

> [http://hackerpublicradio.org/mailman/listinfo/dev\\_hackerpublicradio.org](http://hackerpublicradio.org/mailman/listinfo/dev_hackerpublicradio.org)



From: Dave Morriss <dave.morriss@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 19:43:26 +0100  
Sender: dev-bounces@hackerpublicradio.org  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64;  
rv:16.0) Gecko/20121011 Thunderbird/16.0.1

---

Hi,

I joined the list last week.

So far I have looked through the archives, and assume that some Mumble discussions have taken place at some point.

A few months back I cloned the Gitorious HPR repository and have been adding what work I've done so far to my copy. Having seen Ken's queue.pl script, I wrote some of my own Perl code to apply scheduling rules to a hypothetical queue, and am just in the process of rewriting the concept code in a more robust form.

The thing I'm lacking is an overview of what's being built here. I'm wondering if there's an overview document or similar?

If not, can anyone (singular or plural) explain what the vision is?

I first "met" Ken earlier this year; he asked for suggestions for a script which could build RSS, and I sent him a response. Later, when was in discussion with Ken, I wrote a brief document about what I guessed was going to be the overall design of the Scheduling System. Ken placed it on the Gitorious site. I don't think it was particularly useful, but it might be something that could be built on.

I'd be happy to put together something which encapsulates the plans so far if details could be sent my way. I'm not a Project Manager, and I think this is really Ken's role anyway, but I'm happy to contribute something in this area if it would help.

I know I would appreciate knowing more about the plans at least.

By the way: is this list moderated? Epicanis' message seemed to hang around a while on the Mailman system. Also, what's with the ASCII Art cat stuff?

Dave

--

David Morriss, Scotland, UK | [Dave.Morriss@gmail.com](mailto:Dave.Morriss@gmail.com)

From: Dave Morriss <dave.morriss@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 19:43:26 +0100  
Sender: dev-bounces@hackerpublicradio.org  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64;  
rv:16.0) Gecko/20121011 Thunderbird/16.0.1

---

Hi,

I joined the list last week.

So far I have looked through the archives, and assume that some Mumble discussions have taken place at some point.

A few months back I cloned the Gitorious HPR repository and have been adding what work I've done so far to my copy. Having seen Ken's queue.pl script, I wrote some of my own Perl code to apply scheduling rules to a hypothetical queue, and am just in the process of rewriting the concept code in a more robust form.

The thing I'm lacking is an overview of what's being built here. I'm wondering if there's an overview document or similar?

If not, can anyone (singular or plural) explain what the vision is?

I first "met" Ken earlier this year; he asked for suggestions for a script which could build RSS, and I sent him a response. Later, when was in discussion with Ken, I wrote a brief document about what I guessed was going to be the overall design of the Scheduling System. Ken placed it on the Gitorious site. I don't think it was particularly useful, but it might be something that could be built on.

I'd be happy to put together something which encapsulates the plans so far if details could be sent my way. I'm not a Project Manager, and I think this is really Ken's role anyway, but I'm happy to contribute something in this area if it would help.

I know I would appreciate knowing more about the plans at least.

By the way: is this list moderated? Epicanis' message seemed to hang around a while on the Mailman system. Also, what's with the ASCII Art cat stuff?

Dave

--

David Morriss, Scotland, UK | [Dave.Morriss@gmail.com](mailto:Dave.Morriss@gmail.com)

From: Ken Fallon <ken.fallon@gmail.com>  
To: Dave Morriss <dave.morriss@gmail.com>  
Cc: dev@hackerpublicradio.org  
Subject: Re: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 23:03:09 +0200  
Sender: dev-bounces@hackerpublicradio.org

---

Hi Dave,

For some reason new posters are queued. I only just got a change to change that - remember when I said I was trying to have a quire few months away from hpr :)

Yes we need another mumble session - mostly because of the discussions we had face to face at ogg camp. It should be recoreded and released.

We could do it immediately before the next community news ?

Ken.

On Mon, Oct 15, 2012 at 8:43 PM, Dave Morriss <dave.morriss@gmail.com> wrote:

> Hi,  
>  
> I joined the list last week.  
>  
> So far I have looked through the archives, and assume that some Mumble  
> discussions have taken place at some point.  
>  
> A few months back I cloned the Gitorious HPR repository and have been  
> adding what work I've done so far to my copy. Having seen Ken's queue.pl  
> script, I wrote some of my own Perl code to apply scheduling rules to a  
> hypothetical queue, and am just in the process of rewriting the concept  
> code in a more robust form.  
>  
> The thing I'm lacking is an overview of what's being built here. I'm  
> wondering if there's an overview document or similar?  
>  
> If not, can anyone (singular or plural) explain what the vision is?  
>  
> I first "met" Ken earlier this year; he asked for suggestions for a  
> script which could build RSS, and I sent him a response. Later, when  
> was in discussion with Ken, I wrote a brief document about what I  
> guessed was going to be the overall design of the Scheduling System. Ken  
> placed it on the Gitorious site. I don't think it was particularly  
> useful, but it might be something that could be built on.  
>  
> I'd be happy to put together something which encapsulates the plans so  
> far if details could be sent my way. I'm not a Project Manager, and I  
> think this is really Ken's role anyway, but I'm happy to contribute  
> something in this area if it would help.  
>  
> I know I would appreciate knowing more about the plans at least.  
>  
> By the way: is this list moderated? Epicanis' message seemed to hang  
> around a while on the Mailman system. Also, what's with the ASCII Art  
> cat stuff?  
>  
> Dave  
>  
> --  
> David Morriss, Scotland, UK | [Dave.Morriss@gmail.com](mailto:Dave.Morriss@gmail.com)  
>



From: Ken Fallon <ken.fallon@gmail.com>  
To: Dave Morriss <dave.morriss@gmail.com>  
Cc: dev@hackerpublicradio.org  
Subject: Re: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 23:03:09 +0200  
Sender: dev-bounces@hackerpublicradio.org

---

Hi Dave,

For some reason new posters are queued. I only just got a change to change that - remember when I said I was trying to have a quire few months away from hpr :)

Yes we need another mumble session - mostly because of the discussions we had face to face at ogg camp. It should be recoreded and released.

We could do it immediately before the next community news ?

Ken.

On Mon, Oct 15, 2012 at 8:43 PM, Dave Morriss <dave.morriss@gmail.com> wrote:

> Hi,  
>  
> I joined the list last week.  
>  
> So far I have looked through the archives, and assume that some Mumble  
> discussions have taken place at some point.  
>  
> A few months back I cloned the Gitorious HPR repository and have been  
> adding what work I've done so far to my copy. Having seen Ken's queue.pl  
> script, I wrote some of my own Perl code to apply scheduling rules to a  
> hypothetical queue, and am just in the process of rewriting the concept  
> code in a more robust form.  
>  
> The thing I'm lacking is an overview of what's being built here. I'm  
> wondering if there's an overview document or similar?  
>  
> If not, can anyone (singular or plural) explain what the vision is?  
>  
> I first "met" Ken earlier this year; he asked for suggestions for a  
> script which could build RSS, and I sent him a response. Later, when  
> was in discussion with Ken, I wrote a brief document about what I  
> guessed was going to be the overall design of the Scheduling System. Ken  
> placed it on the Gitorious site. I don't think it was particularly  
> useful, but it might be something that could be built on.  
>  
> I'd be happy to put together something which encapsulates the plans so  
> far if details could be sent my way. I'm not a Project Manager, and I  
> think this is really Ken's role anyway, but I'm happy to contribute  
> something in this area if it would help.  
>  
> I know I would appreciate knowing more about the plans at least.  
>  
> By the way: is this list moderated? Epicanis' message seemed to hang  
> around a while on the Mailman system. Also, what's with the ASCII Art  
> cat stuff?  
>  
> Dave  
>  
> --  
> David Morriss, Scotland, UK | Dave.Morriss@gmail.com  
>



From: Dave Morriss <dave.morriss@gmail.com>  
To: Ken Fallon <ken.fallon@gmail.com>  
Cc: dev@hackerpublicradio.org  
Subject: Re: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 22:19:57 +0100  
Sender: dev-bounces@hackerpublicradio.org  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64;  
rv:16.0) Gecko/20121011 Thunderbird/16.0.1

---

On 15/10/12 22:03, Ken Fallon wrote:  
> For some reason new posters are queued.

In the versions of Mailman I've run (not as current as this one) you can default to member\_moderation so all new members have their moderation flag set. Maybe that's why?

> I only just got a change to change that - remember when I said I was  
> trying to have a quire few months away from hpr :)

Oops! I thought it was \*one\* month!

> Yes we need another mumble session - mostly because of the discussions  
> we had face to face at ogg camp. It should be recoreded and released.  
>  
> We could do it immediately before the next community news ?

I'm up for that. It probably needs some sort of agenda I suggest.

Dave

--

Dave Morriss, Edinburgh, Scotland, UK | [Dave.Morriss@gmail.com](mailto:Dave.Morriss@gmail.com)

From: Dave Morriss <dave.morriss@gmail.com>  
To: Ken Fallon <ken.fallon@gmail.com>  
Cc: dev@hackerpublicradio.org  
Subject: Re: [Dev] Hi Devs!  
Date: Mon, 15 Oct 2012 22:19:57 +0100  
Sender: dev-bounces@hackerpublicradio.org  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64;  
rv:16.0) Gecko/20121011 Thunderbird/16.0.1

---

On 15/10/12 22:03, Ken Fallon wrote:

> For some reason new posters are queued.

In the versions of Mailman I've run (not as current as this one) you can default to member\_moderation so all new members have their moderation flag set. Maybe that's why?

> I only just got a change to change that - remember when I said I was  
> trying to have a quire few months away from hpr :)

Oops! I thought it was \*one\* month!

> Yes we need another mumble session - mostly because of the discussions  
> we had face to face at ogg camp. It should be recoreded and released.  
>  
> We could do it immediately before the next community news ?

I'm up for that. It probably needs some sort of agenda I suggest.

Dave

--

Dave Morriss, Edinburgh, Scotland, UK | [Dave.Morriss@gmail.com](mailto:Dave.Morriss@gmail.com)

From: [epicanis+hpr@dogphilosophy.net](mailto:epicanis+hpr@dogphilosophy.net)  
To: [dev@hackerpublicradio.org](mailto:dev@hackerpublicradio.org)  
Subject: [Dev] Atom feed question(s)  
Date: Mon, 15 Oct 2012 22:48:36 -0400  
Sender: [dev-bounces@hackerpublicradio.org](mailto:dev-bounces@hackerpublicradio.org)  
User-Agent: KMail/4.9.2 (Linux/3.6.2-1-ARCH; KDE/4.9.2; x86\_64; ; )

---

> > and then converts that into a valid individual item to be  
> > incorporated into the Atom feed, is that correct?  
>  
> It should add it to a /Entry element, that is top-level element of a  
> stand-alone Atom Entry Document.  
> <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
[...]  
> The form should be the source of truth and will override all other data  
> with the possible exception of "enclosure" and "type".  
>  
> <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
[...]  
Okay then, let me see if I get the desired process:

The submitter will navigate to a webpage on one server,  
where they'll be greeted with a web form containing fields for  
the various bits of metadata we want, including a textarea for show notes  
to be pasted into, and upload field for the submitted media, and the option  
for one or more image files for upload as well.

The submitter fills in the information, selects the files for upload, and  
pushes the button. Once the data finishes uploading to the receiving script,  
the submitter's work is done (until we possibly add some sort of sanity  
checks, warnings, etc. to the receiving script that might prompt the  
submitter to optionally do something else, do things over, or whatever.  
I don't imagine this in the initial implementation, and maybe never.)

The receiving script takes the metadata, summary, shownotes,  
uploaded audio, and optional uploaded images and moves them all to  
its own directory (with the name format you specified below, and  
located somewhere that the PHP user has write access to but is outside  
the "document root" for the web server and therefore not directly  
accessible by web) after doing some validity checks (e.g. check for  
embedded javascript, possibly check image files, possibly evaluate uploaded  
audio for basic information e.g. duration, format, etc.), and additionally  
generating an appropriately-filled template file for the Atom entry  
(I say "template" as the "enclosure" cannot be filled in until the  
submission has a firm place in the queue and has a known episode  
number [ <http://hackerpublicradio.org/eps/hpr####.ogg|spx|mp3> ],  
right?) The receiving script could also send some sort of alert,  
or will the reception directory just be monitored by someone for  
new uploads?

At that point, the work of the "submission" system is done - the rest  
(audio encoding, checking for intro/outro, human review of metadata,  
setting the encoded audio's place in the queue, etc.) will be initiated by  
a separate system hosted on a different computer.

>  
> > (The former is a more ambitious project to build a web-based system  
> > for generating appropriately-formatted output audio  
[...]  
> We may put links to guide lines where the preference would be flac,  
> wav, ogg etc with bitrates and encoding, but the form should also  
> upload any audio format at all. Even files that it cannot determine the  
> mime type of. The logic is that it is better to have a low fidelity  
> show than no show.

>  
> Album art should not be promoted. The reason is that we reserve fields  
> "Album Art", "Album = Hacker Public Radio", and the first part of  
> "Comment = <http://hackerpublicradio.org>" HPR branding on the version  
> that we distribute. We will probably remove any embedded art as it is  
> know to cause issues with media players. That should not be the case  
> for the versions we upload to Archive.Org.  
> All tagging in the media files will be added from the data submitted in  
> the upload form.

I should have clarified that what I'm talking about there is a system not specifically for HPR use, though if I can get it working and HPR-specific version ought to be pretty simple to build. It's just my "put my metaphorical money where my virtual mouth is" project that I mention towards the end of my latest HPR submission, where I am sick of the complaining that comes from some corners of the internet media world that the reason for not offering legally-free media is that it's "too hard" or "takes too many 'resources'" or whatever. The idea is to have a simple, generic "select the file, push the button, get back finished audio files ready for posting in multiple formats" system.

Where I thought that might come in handy for HPR's process is that (again, assuming I can get it to work right) whoever downloads the original submission audio for processing could potentially use a customized version of the same system with pre-defined metadata (e.g. the HPR "album art", "Album = Hacker Public Radio", "Comment = <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for submissions that don't have it already.

Anyway, that project of mine is focussed entirely on "take input audio, generate finished audio ready for posting somewhere" and doing that one thing as well as my amateur skills can get it to, so it wouldn't be at all relevant to things like scheduling.

>  
> The show notes can be in text, html or xhtml format and the encoding  
> needs to be correct. The show notes may have images supplied so we need  
> to be able to support multiple images.  
>  
> Remember that nothing should be trusted in the upload. We have "hacker"  
> in our name after all. The entire delivery package (media, xml file,  
> image(s) ) should be placed in a non browsable directory, in a  
> subfolder composed of  
> <priority> - the priority of the show  
> \_ - delimiter underscore  
> <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> \_ - delimiter underscore  
> <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
>  
> This folder should not be disclosed to the uploader to prevent people  
> using HPR as a file sharing backup solution.

>  
> For example:  
> 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
[...]

Is the required OSSP uuid executable on the web server such that I can use it via exec(), system(), backtick-operator, etc. or will I need to pull in some other implementation (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?

Also, if the receiving script thinks there's javascript or php or whatever in the description/summary, what should it do? Delete? Strip the seemingly offending section? Sanitize it via e.g. htmlentities();? Or just flag it in some fashion for human review?



From: [epicanis+hpr@dogphilosophy.net](mailto:epicanis+hpr@dogphilosophy.net)  
To: [dev@hackerpublicradio.org](mailto:dev@hackerpublicradio.org)  
Subject: [Dev] Atom feed question(s)  
Date: Mon, 15 Oct 2012 22:48:36 -0400  
Sender: [dev-bounces@hackerpublicradio.org](mailto:dev-bounces@hackerpublicradio.org)  
User-Agent: KMail/4.9.2 (Linux/3.6.2-1-ARCH; KDE/4.9.2; x86\_64; ; )

---

> > and then converts that into a valid individual item to be  
> > incorporated into the Atom feed, is that correct?  
>  
> It should add it to a /Entry element, that is top-level element of a  
> stand-alone Atom Entry Document.  
> <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
[...]  
> The form should be the source of truth and will override all other data  
> with the possible exception of "enclosure" and "type".  
>  
> <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
[...]  
Okay then, let me see if I get the desired process:

The submitter will navigate to a webpage on one server,  
where they'll be greeted with a web form containing fields for  
the various bits of metadata we want, including a textarea for show notes  
to be pasted into, and upload field for the submitted media, and the option  
for one or more image files for upload as well.

The submitter fills in the information, selects the files for upload, and  
pushes the button. Once the data finishes uploading to the receiving script,  
the submitter's work is done (until we possibly add some sort of sanity  
checks, warnings, etc. to the receiving script that might prompt the  
submitter to optionally do something else, do things over, or whatever.  
I don't imagine this in the initial implementation, and maybe never.)

The receiving script takes the metadata, summary, shownotes,  
uploaded audio, and optional uploaded images and moves them all to  
its own directory (with the name format you specified below, and  
located somewhere that the PHP user has write access to but is outside  
the "document root" for the web server and therefore not directly  
accessible by web) after doing some validity checks (e.g. check for  
embedded javascript, possibly check image files, possibly evaluate uploaded  
audio for basic information e.g. duration, format, etc.), and additionally  
generating an appropriately-filled template file for the Atom entry  
(I say "template" as the "enclosure" cannot be filled in until the  
submission has a firm place in the queue and has a known episode  
number [ <http://hackerpublicradio.org/eps/hpr####.ogg|spx|mp3> ],  
right?) The receiving script could also send some sort of alert,  
or will the reception directory just be monitored by someone for  
new uploads?

At that point, the work of the "submission" system is done - the rest  
(audio encoding, checking for intro/outro, human review of metadata,  
setting the encoded audio's place in the queue, etc.) will be initiated by  
a separate system hosted on a different computer.

>  
> > (The former is a more ambitious project to build a web-based system  
> > for generating appropriately-formatted output audio  
[...]  
> We may put links to guide lines where the preference would be flac,  
> wav, ogg etc with bitrates and encoding, but the form should also  
> upload any audio format at all. Even files that it cannot determine the  
> mime type of. The logic is that it is better to have a low fidelity  
> show than no show.

>  
> Album art should not be promoted. The reason is that we reserve fields  
> "Album Art", "Album = Hacker Public Radio", and the first part of  
> "Comment = <http://hackerpublicradio.org>" HPR branding on the version  
> that we distribute. We will probably remove any embedded art as it is  
> know to cause issues with media players. That should not be the case  
> for the versions we upload to Archive.Org.  
> All tagging in the media files will be added from the data submitted in  
> the upload form.

I should have clarified that what I'm talking about there is a system not specifically for HPR use, though if I can get it working and HPR-specific version ought to be pretty simple to build. It's just my "put my metaphorical money where my virtual mouth is" project that I mention towards the end of my latest HPR submission, where I am sick of the complaining that comes from some corners of the internet media world that the reason for not offering legally-free media is that it's "too hard" or "takes too many 'resources'" or whatever. The idea is to have a simple, generic "select the file, push the button, get back finished audio files ready for posting in multiple formats" system.

Where I thought that might come in handy for HPR's process is that (again, assuming I can get it to work right) whoever downloads the original submission audio for processing could potentially use a customized version of the same system with pre-defined metadata (e.g. the HPR "album art", "Album = Hacker Public Radio", "Comment = <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for submissions that don't have it already.

Anyway, that project of mine is focussed entirely on "take input audio, generate finished audio ready for posting somewhere" and doing that one thing as well as my amateur skills can get it to, so it wouldn't be at all relevant to things like scheduling.

>  
> The show notes can be in text, html or xhtml format and the encoding  
> needs to be correct. The show notes may have images supplied so we need  
> to be able to support multiple images.  
>  
> Remember that nothing should be trusted in the upload. We have "hacker"  
> in our name after all. The entire delivery package (media, xml file,  
> image(s) ) should be placed in a non browsable directory, in a  
> subfolder composed of  
> <priority> - the priority of the show  
> \_ - delimiter underscore  
> <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> \_ - delimiter underscore  
> <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
>  
> This folder should not be disclosed to the uploader to prevent people  
> using HPR as a file sharing backup solution.

> For example:  
> 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
[...]

Is the required OSSP uuid executable on the web server such that I can use it via exec(), system(), backtick-operator, etc. or will I need to pull in some other implementation (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?

Also, if the receiving script thinks there's javascript or php or whatever in the description/summary, what should it do? Delete? Strip the seemingly offending section? Sanitize it via e.g. `htmlentities()`? Or just flag it in some fashion for human review?



From: [epicanis+hpr@dogphilosophy.net](mailto:epicanis+hpr@dogphilosophy.net)  
To: [dev@hackerpublicradio.org](mailto:dev@hackerpublicradio.org)  
Subject: [Dev] Atom feed question(s)  
Date: Mon, 15 Oct 2012 22:48:36 -0400  
Sender: [dev-bounces@hackerpublicradio.org](mailto:dev-bounces@hackerpublicradio.org)  
User-Agent: KMail/4.9.2 (Linux/3.6.2-1-ARCH; KDE/4.9.2; x86\_64; ; )

---

> > and then converts that into a valid individual item to be  
> > incorporated into the Atom feed, is that correct?  
>  
> It should add it to a /Entry element, that is top-level element of a  
> stand-alone Atom Entry Document.  
> <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
[...]  
> The form should be the source of truth and will override all other data  
> with the possible exception of "enclosure" and "type".  
>  
> <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
[...]  
Okay then, let me see if I get the desired process:

The submitter will navigate to a webpage on one server,  
where they'll be greeted with a web form containing fields for  
the various bits of metadata we want, including a textarea for show notes  
to be pasted into, and upload field for the submitted media, and the option  
for one or more image files for upload as well.

The submitter fills in the information, selects the files for upload, and  
pushes the button. Once the data finishes uploading to the receiving script,  
the submitter's work is done (until we possibly add some sort of sanity  
checks, warnings, etc. to the receiving script that might prompt the  
submitter to optionally do something else, do things over, or whatever.  
I don't imagine this in the initial implementation, and maybe never.)

The receiving script takes the metadata, summary, shownotes,  
uploaded audio, and optional uploaded images and moves them all to  
its own directory (with the name format you specified below, and  
located somewhere that the PHP user has write access to but is outside  
the "document root" for the web server and therefore not directly  
accessible by web) after doing some validity checks (e.g. check for  
embedded javascript, possibly check image files, possibly evaluate uploaded  
audio for basic information e.g. duration, format, etc.), and additionally  
generating an appropriately-filled template file for the Atom entry  
(I say "template" as the "enclosure" cannot be filled in until the  
submission has a firm place in the queue and has a known episode  
number [ <http://hackerpublicradio.org/eps/hpr####.ogg|spx|mp3> ],  
right?) The receiving script could also send some sort of alert,  
or will the reception directory just be monitored by someone for  
new uploads?

At that point, the work of the "submission" system is done - the rest  
(audio encoding, checking for intro/outro, human review of metadata,  
setting the encoded audio's place in the queue, etc.) will be initiated by  
a separate system hosted on a different computer.

>  
> > (The former is a more ambitious project to build a web-based system  
> > for generating appropriately-formatted output audio  
[...]  
> We may put links to guide lines where the preference would be flac,  
> wav, ogg etc with bitrates and encoding, but the form should also  
> upload any audio format at all. Even files that it cannot determine the  
> mime type of. The logic is that it is better to have a low fidelity  
> show than no show.

>  
> Album art should not be promoted. The reason is that we reserve fields  
> "Album Art", "Album = Hacker Public Radio", and the first part of  
> "Comment = <http://hackerpublicradio.org>" HPR branding on the version  
> that we distribute. We will probably remove any embedded art as it is  
> know to cause issues with media players. That should not be the case  
> for the versions we upload to Archive.Org.  
> All tagging in the media files will be added from the data submitted in  
> the upload form.

I should have clarified that what I'm talking about there is a system not specifically for HPR use, though if I can get it working and HPR-specific version ought to be pretty simple to build. It's just my "put my metaphorical money where my virtual mouth is" project that I mention towards the end of my latest HPR submission, where I am sick of the complaining that comes from some corners of the internet media world that the reason for not offering legally-free media is that it's "too hard" or "takes too many 'resources'" or whatever. The idea is to have a simple, generic "select the file, push the button, get back finished audio files ready for posting in multiple formats" system.

Where I thought that might come in handy for HPR's process is that (again, assuming I can get it to work right) whoever downloads the original submission audio for processing could potentially use a customized version of the same system with pre-defined metadata (e.g. the HPR "album art", "Album = Hacker Public Radio", "Comment = <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for submissions that don't have it already.

Anyway, that project of mine is focussed entirely on "take input audio, generate finished audio ready for posting somewhere" and doing that one thing as well as my amateur skills can get it to, so it wouldn't be at all relevant to things like scheduling.

>  
> The show notes can be in text, html or xhtml format and the encoding  
> needs to be correct. The show notes may have images supplied so we need  
> to be able to support multiple images.  
>  
> Remember that nothing should be trusted in the upload. We have "hacker"  
> in our name after all. The entire delivery package (media, xml file,  
> image(s) ) should be placed in a non browsable directory, in a  
> subfolder composed of  
> <priority> - the priority of the show  
> \_ - delimiter underscore  
> <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> \_ - delimiter underscore  
> <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
>  
> This folder should not be disclosed to the uploader to prevent people  
> using HPR as a file sharing backup solution.

> For example:  
> 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
[...]

Is the required OSSP uuid executable on the web server such that I can use it via exec(), system(), backtick-operator, etc. or will I need to pull in some other implementation (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?

Also, if the receiving script thinks there's javascript or php or whatever in the description/summary, what should it do? Delete? Strip the seemingly offending section? Sanitize it via e.g. htmlentities();? Or just flag it in some fashion for human review?



From: Ken Fallon <ken.fallon@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: Re: [Dev] Atom feed question(s)  
Date: Tue, 16 Oct 2012 06:58:43 +0200  
X-Mailer: Claws Mail 3.8.1 (GTK+ 2.24.11; x86\_64-redhat-linux-gnu)

---

On Mon, 15 Oct 2012 22:48:36 -0400  
epicanis+hpr@dogphilosophy.net wrote:

> > > and then converts that into a valid individual item to be  
> > > incorporated into the Atom feed, is that correct?  
> >  
> > It should add it to a /Entry element, that is top-level element of a  
> > stand-alone Atom Entry Document.  
> > <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
> > [...]   
> > The form should be the source of truth and will override all other  
> > data with the possible exception of "enclosure" and "type".  
> >  
> > <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
> > [...]   
> > Okay then, let me see if I get the desired process:  
> >  
> > The submitter will navigate to a webpage on one server,  
> > where they'll be greeted with a web form containing fields for  
> > the various bits of metadata we want, including a textarea for show  
> > notes to be pasted into, and upload field for the submitted media,  
> > and the option for one or more image files for upload as well.  
> > Yes exactly.

> > The submitter fills in the information, selects the files for upload,  
> > and pushes the button. Once the data finishes uploading to the  
> > receiving script, the submitter's work is done (until we possibly add  
> > some sort of sanity checks, warnings, etc. to the receiving script  
> > that might prompt the submitter to optionally do something else, do  
> > things over, or whatever. I don't imagine this in the initial  
> > implementation, and maybe never.)  
> >  
> > The receiving script takes the metadata, summary, shownotes,  
> > uploaded audio, and optional uploaded images and moves them all to  
> > its own directory (with the name format you specified below, and  
> > located somewhere that the PHP user has write access to but is outside  
> > the "document root" for the web server and therefore not directly  
> > accessible by web)

Actually no. That's the job of the upload script.

> > after doing some validity checks (e.g. check for  
> > embedded javascript, possibly check image files, possibly evaluate  
> > uploaded audio for basic information e.g. duration, format, etc.),  
> > and additionally generating an appropriately-filled template file for  
> > the Atom entry (I say "template" as the "enclosure" cannot be filled  
> > in until the submission has a firm place in the queue and has a known  
> > episode number [ <http://hackerpublicradio.org/eps/hpr####.ogg|spx|mp3>  
> > ], right?)

Again this is part of the upload script running on the frontend web  
server. You might decide to have a client part to the form running  
javascript and a server part running php but the line of demarcation is  
that a folder is created and in there is the "show" containing the  
media file, shownotes, and optional images.

> > The receiving script could also send some sort of alert,  
> > or will the reception directory just be monitored by someone for

> new uploads?  
> At that point, the work of the "submission" system is done - the rest  
> (audio encoding, checking for intro/outro, human review of metadata,  
> setting the encoded audio's place in the queue, etc.) will be  
> initiated by a separate system hosted on a different computer.

Yes the backend script will periodically check for new folders and when the upload is finished. (We need a way to flag that - possibly renaming from

"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af.temp" to  
"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af"

At this point that directory probably contains a show but it could easily contain porn, viruses or any other stuff someone decides to send us.

> >  
> > > (The former is a more ambitious project to build a web-based  
> > > system for generating appropriately-formatted output audio  
> > [...]  
> > We may put links to guide lines where the preference would be flac,  
> > wav, ogg etc with bitrates and encoding, but the form should also  
> > upload any audio format at all. Even files that it cannot determine  
> > the mime type of. The logic is that it is better to have a low  
> > fidelity show than no show.

> >  
> > Album art should not be promoted. The reason is that we reserve  
> > fields "Album Art", "Album = Hacker Public Radio", and the first  
> > part of "Comment = <http://hackerpublicradio.org>" HPR branding on  
> > the version that we distribute. We will probably remove any  
> > embedded art as it is know to cause issues with media players. That  
> > should not be the case for the versions we upload to Archive.Org.  
> > All tagging in the media files will be added from the data submitted  
> > in the upload form.

>  
> I should have clarified that what I'm talking about there is a system  
> not specifically for HPR use, though if I can get it working and  
> HPR-specific version ought to be pretty simple to build. It's just my  
> "put my metaphorical money where my virtual mouth is" project that I  
> mention towards the end of my latest HPR submission, where I am sick  
> of the complaining that comes from some corners of the internet media  
> world that the reason for not offering legally-free media is that  
> it's "too hard" or "takes too many 'resources'" or whatever. The idea  
> is to have a simple, generic "select the file, push the button, get  
> back finished audio files ready for posting in multiple formats"  
> system.

Yes Yes Yes. A non HPR system is ideal. Keep that thought. Not a lot of point writing something that is HPR specific. I need to listen to your show.

I also toyed with a "record now" button on the site but my sights have been lowered somewhat.

>  
> Where I thought that might come in handy for HPR's process is that  
> (again, assuming I can get it to work right) whoever downloads the  
> original submission audio for processing could potentially use a  
> customized version of the same system with pre-defined metadata (e.g.  
> the HPR "album art", "Album = Hacker Public Radio", "Comment =  
> <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for  
> submissions that don't have it already.

Nothing saying the can't do that and in fact that's the whole point.  
It's just from a security point of view, it makes sense to separate both processes.

>  
> Anyway, that project of mine is focussed entirely on "take input  
> audio, generate finished audio ready for posting somewhere" and doing  
> that one thing as well as my amateur skills can get it to, so it  
> wouldn't be at all relevant to things like scheduling.  
But it covers two of the important sections

Acquisition  
Transcoding  
Scheduling

>  
> >  
> > The show notes can be in text, html or xhtml format and the encoding  
> > needs to be correct. The show notes may have images supplied so we  
> > need to be able to support multiple images.  
> >  
> > Remember that nothing should be trusted in the upload. We have  
> > "hacker" in our name after all. The entire delivery package (media,  
> > xml file, image(s) ) should be placed in a non browsable directory,  
> > in a subfolder composed of  
> > <priority> - the priority of the show  
> > \_ - delimiter underscore  
> > <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> > \_ - delimiter underscore  
> > <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
> >  
> > This folder should not be disclosed to the uploader to prevent  
> > people using HPR as a file sharing backup solution.  
> >  
> > For example:  
> > 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
> > [...]  
> > Is the required OSSP uuid executable on the web server such that I can  
> > use it via exec(), system(), backtick-operator, etc. or will I need  
> > to pull in some other implementation  
> > (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?  
Pull in what you like, that was just for example only.

>  
> Also, if the receiving script thinks there's javascript or php  
> or whatever in the description/summary, what should it do?  
> Delete? Strip the seemingly offending section? Sanitize it  
> via e.g. htmlentities();? Or just flag it in some fashion for  
> human review?

>  
Hummmm need to think about that. If you detect it at input, then inform  
them. If they are using some UTF-8 hack to get them in then just accept  
it and put it in the "h\_" for hacker queue :) Don't do anything to  
alert them to the fact that you noticed it. We'll check it manually and  
if it's a genuine show, they will be informed manually. If not a time  
out will take place.

By the way the backend will send an email out to the address in the  
file to confirm the show.

From: Ken Fallon <ken.fallon@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: Re: [Dev] Atom feed question(s)  
Date: Tue, 16 Oct 2012 06:58:43 +0200  
Sender: dev-bounces@hackerpublicradio.org  
X-Mailer: Claws Mail 3.8.1 (GTK+ 2.24.11; x86\_64-redhat-linux-gnu)

---

On Mon, 15 Oct 2012 22:48:36 -0400  
epicanis+hpr@dogphilosophy.net wrote:

> > > and then converts that into a valid individual item to be  
> > > incorporated into the Atom feed, is that correct?  
> >  
> > It should add it to a /Entry element, that is top-level element of a  
> > stand-alone Atom Entry Document.  
> > <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
> > [...]   
> > The form should be the source of truth and will override all other  
> > data with the possible exception of "enclosure" and "type".  
> >  
> > <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
> > [...]   
> > Okay then, let me see if I get the desired process:  
> >  
> > The submitter will navigate to a webpage on one server,  
> > where they'll be greeted with a web form containing fields for  
> > the various bits of metadata we want, including a textarea for show  
> > notes to be pasted into, and upload field for the submitted media,  
> > and the option for one or more image files for upload as well.  
Yes exactly.

> > The submitter fills in the information, selects the files for upload,  
> > and pushes the button. Once the data finishes uploading to the  
> > receiving script, the submitter's work is done (until we possibly add  
> > some sort of sanity checks, warnings, etc. to the receiving script  
> > that might prompt the submitter to optionally do something else, do  
> > things over, or whatever. I don't imagine this in the initial  
> > implementation, and maybe never.)  
> >  
> > The receiving script takes the metadata, summary, shownotes,  
> > uploaded audio, and optional uploaded images and moves them all to  
> > its own directory (with the name format you specified below, and  
> > located somewhere that the PHP user has write access to but is outside  
> > the "document root" for the web server and therefore not directly  
> > accessible by web)

Actually no. That's the job of the upload script.

> > after doing some validity checks (e.g. check for  
> > embedded javascript, possibly check image files, possibly evaluate  
> > uploaded audio for basic information e.g. duration, format, etc.),  
> > and additionally generating an appropriately-filled template file for  
> > the Atom entry (I say "template" as the "enclosure" cannot be filled  
> > in until the submission has a firm place in the queue and has a known  
> > episode number [ <http://hackerpublicradio.org/eps/hpr#####.ogg|spx|mp3>  
> > ], right?)

Again this is part of the upload script running on the frontend web  
server. You might decide to have a client part to the form running  
javascript and a server part running php but the line of demarcation is  
that a folder is created and in there is the "show" containing the  
media file, shownotes, and optional images.

> > The receiving script could also send some sort of alert,

> or will the reception directory just be monitored by someone for  
> new uploads?  
> At that point, the work of the "submission" system is done - the rest  
> (audio encoding, checking for intro/outro, human review of metadata,  
> setting the encoded audio's place in the queue, etc.) will be  
> initiated by a separate system hosted on a different computer.

Yes the backend script will periodically check for new folders and  
when the upload is finished. (We need a way to flag that - possibly  
renaming from

"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af.temp" to  
"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af"

At this point that directory probably contains a show but it could  
easily contain porn, viruses or any other stuff someone decides to send  
us.

> >  
> > > (The former is a more ambitious project to build a web-based  
> > > system for generating appropriately-formatted output audio  
> > [...]  
> > We may put links to guide lines where the preference would be flac,  
> > wav, ogg etc with bitrates and encoding, but the form should also  
> > upload any audio format at all. Even files that it cannot determine  
> > the mime type of. The logic is that it is better to have a low  
> > fidelity show than no show.

> >  
> > Album art should not be promoted. The reason is that we reserve  
> > fields "Album Art", "Album = Hacker Public Radio", and the first  
> > part of "Comment = <http://hackerpublicradio.org>" HPR branding on  
> > the version that we distribute. We will probably remove any  
> > embedded art as it is know to cause issues with media players. That  
> > should not be the case for the versions we upload to Archive.Org.  
> > All tagging in the media files will be added from the data submitted  
> > in the upload form.

>  
> I should have clarified that what I'm talking about there is a system  
> not specifically for HPR use, though if I can get it working and  
> HPR-specific version ought to be pretty simple to build. It's just my  
> "put my metaphorical money where my virtual mouth is" project that I  
> mention towards the end of my latest HPR submission, where I am sick  
> of the complaining that comes from some corners of the internet media  
> world that the reason for not offering legally-free media is that  
> it's "too hard" or "takes too many 'resources'" or whatever. The idea  
> is to have a simple, generic "select the file, push the button, get  
> back finished audio files ready for posting in multiple formats"  
> system.

Yes Yes Yes. A non HPR system is ideal. Keep that thought. Not a lot of point  
writing something that is HPR specific. I need to listen to your show.

I also toyed with a "record now" button on the site but my sights have  
been lowered somewhat.

>  
> Where I thought that might come in handy for HPR's process is that  
> (again, assuming I can get it to work right) whoever downloads the  
> original submission audio for processing could potentially use a  
> customized version of the same system with pre-defined metadata (e.g.  
> the HPR "album art", "Album = Hacker Public Radio", "Comment =  
> <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for  
> submissions that don't have it already.

Nothing saying the can't do that and in fact that's the whole point.  
It's just from a security point of view, it makes sense to separate both  
processes.

>  
> Anyway, that project of mine is focussed entirely on "take input  
> audio, generate finished audio ready for posting somewhere" and doing  
> that one thing as well as my amateur skills can get it to, so it  
> wouldn't be at all relevant to things like scheduling.  
But it covers two of the important sections

Acquisition  
Transcoding  
Scheduling

>  
> >  
> > The show notes can be in text, html or xhtml format and the encoding  
> > needs to be correct. The show notes may have images supplied so we  
> > need to be able to support multiple images.  
> >  
> > Remember that nothing should be trusted in the upload. We have  
> > "hacker" in our name after all. The entire delivery package (media,  
> > xml file, image(s) ) should be placed in a non browsable directory,  
> > in a subfolder composed of  
> > <priority> - the priority of the show  
> > \_ - delimiter underscore  
> > <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> > \_ - delimiter underscore  
> > <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
> >  
> > This folder should not be disclosed to the uploader to prevent  
> > people using HPR as a file sharing backup solution.  
> >  
> > For example:  
> > 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
> > [...]  
> > Is the required OSSP uuid executable on the web server such that I can  
> > use it via exec(), system(), backtick-operator, etc. or will I need  
> > to pull in some other implementation  
> > (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?  
Pull in what you like, that was just for example only.

>  
> Also, if the receiving script thinks there's javascript or php  
> or whatever in the description/summary, what should it do?  
> Delete? Strip the seemingly offending section? Sanitize it  
> via e.g. htmlentities();? Or just flag it in some fashion for  
> human review?

>  
Hummmm need to think about that. If you detect it at input, then inform  
them. If they are using some UTF-8 hack to get them in then just accept  
it and put it in the "h\_" for hacker queue :) Don't do anything to  
alert them to the fact that you noticed it. We'll check it manually and  
if it's a genuine show, they will be informed manually. If not a time  
out will take place.

By the way the backend will send an email out to the address in the  
file to confirm the show.

From: Ken Fallon <ken.fallon@gmail.com>  
To: dev@hackerpublicradio.org  
Subject: Re: [Dev] Atom feed question(s)  
Date: Tue, 16 Oct 2012 06:58:43 +0200  
Sender: dev-bounces@hackerpublicradio.org  
X-Mailer: Claws Mail 3.8.1 (GTK+ 2.24.11; x86\_64-redhat-linux-gnu)

---

On Mon, 15 Oct 2012 22:48:36 -0400  
epicanis+hpr@dogphilosophy.net wrote:

> > > and then converts that into a valid individual item to be  
> > > incorporated into the Atom feed, is that correct?  
> >  
> > It should add it to a /Entry element, that is top-level element of a  
> > stand-alone Atom Entry Document.  
> > <https://tools.ietf.org/html/rfc4287#section-4.1.2>  
> > [...]  
> > The form should be the source of truth and will override all other  
> > data with the possible exception of "enclosure" and "type".  
> >  
> > <https://tools.ietf.org/html/rfc4287#section-4.2.7>  
> > [...]  
> Okay then, let me see if I get the desired process:  
>  
> The submitter will navigate to a webpage on one server,  
> where they'll be greeted with a web form containing fields for  
> the various bits of metadata we want, including a textarea for show  
> notes to be pasted into, and upload field for the submitted media,  
> and the option for one or more image files for upload as well.  
Yes exactly.

> The submitter fills in the information, selects the files for upload,  
> and pushes the button. Once the data finishes uploading to the  
> receiving script, the submitter's work is done (until we possibly add  
> some sort of sanity checks, warnings, etc. to the receiving script  
> that might prompt the submitter to optionally do something else, do  
> things over, or whatever. I don't imagine this in the initial  
> implementation, and maybe never.)  
>  
> The receiving script takes the metadata, summary, shownotes,  
> uploaded audio, and optional uploaded images and moves them all to  
> its own directory (with the name format you specified below, and  
> located somewhere that the PHP user has write access to but is outside  
> the "document root" for the web server and therefore not directly  
> accessible by web)

Actually no. That's the job of the upload script.

> after doing some validity checks (e.g. check for  
> embedded javascript, possibly check image files, possibly evaluate  
> uploaded audio for basic information e.g. duration, format, etc.),  
> and additionally generating an appropriately-filled template file for  
> the Atom entry (I say "template" as the "enclosure" cannot be filled  
> in until the submission has a firm place in the queue and has a known  
> episode number [ <http://hackerpublicradio.org/eps/hpr#####.ogg|spx|mp3>  
> ], right?)

Again this is part of the upload script running on the frontend web  
server. You might decide to have a client part to the form running  
javascript and a server part running php but the line of demarcation is  
that a folder is created and in there is the "show" containing the  
media file, shownotes, and optional images.

> The receiving script could also send some sort of alert,

> or will the reception directory just be monitored by someone for  
> new uploads?  
> At that point, the work of the "submission" system is done - the rest  
> (audio encoding, checking for intro/outro, human review of metadata,  
> setting the encoded audio's place in the queue, etc.) will be  
> initiated by a separate system hosted on a different computer.

Yes the backend script will periodically check for new folders and  
when the upload is finished. (We need a way to flag that - possibly  
renaming from

"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af.temp" to  
"1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af"

At this point that directory probably contains a show but it could  
easily contain porn, viruses or any other stuff someone decides to send  
us.

> >  
> > > (The former is a more ambitious project to build a web-based  
> > > system for generating appropriately-formatted output audio  
> > [...]  
> > We may put links to guide lines where the preference would be flac,  
> > wav, ogg etc with bitrates and encoding, but the form should also  
> > upload any audio format at all. Even files that it cannot determine  
> > the mime type of. The logic is that it is better to have a low  
> > fidelity show than no show.

> >  
> > Album art should not be promoted. The reason is that we reserve  
> > fields "Album Art", "Album = Hacker Public Radio", and the first  
> > part of "Comment = <http://hackerpublicradio.org>" HPR branding on  
> > the version that we distribute. We will probably remove any  
> > embedded art as it is know to cause issues with media players. That  
> > should not be the case for the versions we upload to Archive.Org.  
> > All tagging in the media files will be added from the data submitted  
> > in the upload form.

>  
> I should have clarified that what I'm talking about there is a system  
> not specifically for HPR use, though if I can get it working and  
> HPR-specific version ought to be pretty simple to build. It's just my  
> "put my metaphorical money where my virtual mouth is" project that I  
> mention towards the end of my latest HPR submission, where I am sick  
> of the complaining that comes from some corners of the internet media  
> world that the reason for not offering legally-free media is that  
> it's "too hard" or "takes too many 'resources'" or whatever. The idea  
> is to have a simple, generic "select the file, push the button, get  
> back finished audio files ready for posting in multiple formats"  
> system.

Yes Yes Yes. A non HPR system is ideal. Keep that thought. Not a lot of point  
writing something that is HPR specific. I need to listen to your show.

I also toyed with a "record now" button on the site but my sights have  
been lowered somewhat.

>  
> Where I thought that might come in handy for HPR's process is that  
> (again, assuming I can get it to work right) whoever downloads the  
> original submission audio for processing could potentially use a  
> customized version of the same system with pre-defined metadata (e.g.  
> the HPR "album art", "Album = Hacker Public Radio", "Comment =  
> <http://hackerpublicradio.org>") and a "add intro/outro?" checkbox for  
> submissions that don't have it already.

Nothing saying the can't do that and in fact that's the whole point.  
It's just from a security point of view, it makes sense to separate both  
processes.

>  
> Anyway, that project of mine is focussed entirely on "take input  
> audio, generate finished audio ready for posting somewhere" and doing  
> that one thing as well as my amateur skills can get it to, so it  
> wouldn't be at all relevant to things like scheduling.  
But it covers two of the important sections

Acquisition  
Transcoding  
Scheduling

>  
> >  
> > The show notes can be in text, html or xhtml format and the encoding  
> > needs to be correct. The show notes may have images supplied so we  
> > need to be able to support multiple images.  
> >  
> > Remember that nothing should be trusted in the upload. We have  
> > "hacker" in our name after all. The entire delivery package (media,  
> > xml file, image(s) ) should be placed in a non browsable directory,  
> > in a subfolder composed of  
> > <priority> - the priority of the show  
> > \_ - delimiter underscore  
> > <date stamp> - in the format "date -u +%Y-%m-%dT%H-%M-%SZ"  
> > \_ - delimiter underscore  
> > <uuid> OSSP uuid - Universally Unique Identifier Command-Line Tool  
> >  
> > This folder should not be disclosed to the uploader to prevent  
> > people using HPR as a file sharing backup solution.  
> >  
> > For example:  
> > 1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af  
> > [...]  
> > Is the required OSSP uuid executable on the web server such that I can  
> > use it via exec(), system(), backtick-operator, etc. or will I need  
> > to pull in some other implementation  
> > (e.g. <https://github.com/fredriklinberg/class.uuid.php>)?  
Pull in what you like, that was just for example only.

>  
> > Also, if the receiving script thinks there's javascript or php  
> > or whatever in the description/summary, what should it do?  
> > Delete? Strip the seemingly offending section? Sanitize it  
> > via e.g. htmlentities();? Or just flag it in some fashion for  
> > human review?  
>

Hummmm need to think about that. If you detect it at input, then inform them. If they are using some UTF-8 hack to get them in then just accept it and put it in the "h\_" for hacker queue :) Don't do anything to alert them to the fact that you noticed it. We'll check it manually and if it's a genuine show, they will be informed manually. If not a time out will take place.

By the way the backend will send an email out to the address in the file to confirm the show.

From: [epicanis+hpr@dogphilosophy.net](mailto:epicanis+hpr@dogphilosophy.net)  
To: [dev@hackerpublicradio.org](mailto:dev@hackerpublicradio.org)  
Subject: Re: [Dev] Atom feed question(s)  
Date: Tue, 16 Oct 2012 16:22:30 -0400  
Sender: [dev-bounces@hackerpublicradio.org](mailto:dev-bounces@hackerpublicradio.org)  
User-Agent: KMail/4.9.2 (Linux/3.6.2-1-ARCH; KDE/4.9.2; x86\_64; ; )

---

[...]  
> > The receiving script takes the metadata, summary, shownotes,  
> > uploaded audio, and optional uploaded images and moves them all to  
> > its own directory [...]  
> >  
> Actually no. That's the job of the upload script.  
>  
> > after doing some validity checks (e.g. check for  
> > embedded javascript, possibly check image files, possibly evaluate  
> > uploaded audio for basic information e.g. duration, format, etc.),  
> > and additionally generating an appropriately-filled template file for  
> > the Atom entry (I say "template" as the "enclosure" cannot be filled  
> > in until the submission has a firm place in the queue and has a known  
> > episode number [ <http://hackerpublicradio.org/eps/hpr####.ogg|spx|mp3>  
> > ], right?)  
>  
> Again this is part of the upload script running on the frontend web  
> server. You might decide to have a client part to the form running  
> javascript and a server part running php but the line of demarcation is  
> that a folder is created and in there is the "show" containing the  
> media file, shownotes, and optional images.

I should clarify - I'm thinking of it as the "receiving script" because  
it's the PHP script that the uploaded file is being sent to from the  
submission form.

(I'm imagining it as two files, a .html or .php file with the initial  
submission form, and a separate .php "receiving script" file that is the  
"action=" target of the form. This could be done as a single file if that's  
preferable though, i.e. a  
"if(\$\_REQUEST['stuff']) { //receive file } else { //print submission form }"  
sort of thing.

[...]  
>  
> Yes the backend script will periodically check for new folders and  
> when the upload is finished. (We need a way to flag that - possibly  
> renaming from  
> "1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af.temp" to  
> "1\_2012-10-15\_19-44-20Z\_fdcd9992-16ef-11e2-8bbd-d4bed91e28af"  
>  
> At this point that directory probably contains a show but it could  
> easily contain porn, viruses or any other stuff someone decides to send  
> us.  
>  
[...slightly out of order repasting here...]  
>  
> By the way the backend will send an email out to the address in the  
> file to confirm the show.

Okay, good, so I can just concentrate on the "take the input data and files  
and stick them in the appropriate directory" task. How soon will  
HPR be likely to switch to Atom feeds? (i.e. would it be useful to  
have this sooner, but have it generate old-school RSS template for  
the existing feed in the meantime?)

[...]  
> Yes Yes Yes. A non HPR system is ideal. Keep that thought. Not a lot of

> point  
> writing something that is HPR specific. I need to listen to your show.  
>  
> I also toyed with a "record now" button on the site but my sights have  
> been lowered somewhat.

>  
Looks like my latest submission should be hitting the feed late next week,  
so not long now! It should also make clear that I firmly approve of what the  
hpr "syndication" page currently looks like, incidentally (other than maybe  
not having an opus feed option :-)

Assuming there's interest (I'm guessing there will be) I was planning to do a  
more technical "how to produce legally-free media" followup episode. It'd  
probably make sense to include some information about how I'm trying to  
approach the "web-based encoder" thing while I'm at it.

I wouldn't necessarily give up on a "record now" button, though it's  
not necessarily feasible right now. I would have sworn there was  
an upcoming standard for access to local microphones and webcams  
due to hit at least Firefox in the relatively near future.

> > Where I thought that might come in handy for HPR's process is that  
> > (again, assuming I can get it to work right) whoever downloads the  
> > original submission audio for processing could potentially use a  
> > customized version of the same system [...]  
>  
> Nothing saying the can't do that and in fact that's the whole point.  
> It's just from a security point of view, it makes sense to separate both  
> processes.

I agree, especially while there's a naturally good place for human  
intervention between the two stages.

>  
> > Anyway, that project of mine is focussed entirely on "take input  
> > audio, generate finished audio ready for posting somewhere" and doing  
> > that one thing as well as my amateur skills can get it to, so it  
> > wouldn't be at all relevant to things like scheduling.

> But it covers two of the important sections

>  
> Acquisition  
> Transcoding  
> Scheduling

[...]

> > Is the required OSSP uuid executable on the web server[...]

> Pull in what you like, that was just for example only.

No problem - just wasn't sure if you'd "standardized on" the OSSP  
implementation. I've got no problem doing it either way, though I  
tend to lean towards minimizing external dependencies when I can.

Though if there's a copy of ffmpeg executable by PHP on the server  
I could have the script pre-report file duration and format and such...

> > Also, if the receiving script thinks there's javascript or php  
> > or whatever in the description/summary, what should it do?  
> > Delete? Strip the seemingly offending section? Sanitize it  
> > via e.g. htmlentities();? Or just flag it in some fashion for  
> > human review?

>  
> Hummmm need to think about that. If you detect it at input, then inform  
> them. If they are using some UTF-8 hack to get them in then just accept

> it and put it in the "h\_" for hacker queue :) Don't do anything to  
> alert them to the fact that you noticed it. We'll check it manually and  
> if it's a genuine show, they will be informed manually. If not a time  
> out will take place.

Okay, so I'll have some sort of (ironically enough) basic javascript-based checker for javascript in the text on the submitter input form, and a PHP-based, somewhat harsher checker on the "receiving script" that will flag possibly-suspicious-looking uploads by prefixing the priority-level with an "h" at the beginning of the directory name (right?).

From: Ken Fallon <ken.fallon@gmail.com>  
To: epicanis+hpr@dogphilosophy.net  
Cc: dev@hackerpublicradio.org  
Subject: Re: [Dev] Atom feed question(s)  
Date: Thu, 18 Oct 2012 21:52:17 +0200  
Sender: dev-bounces@hackerpublicradio.org

---

Sorry I took so long to reply.

On Tue, Oct 16, 2012 at 10:22 PM, <epicanis+hpr@dogphilosophy.net> wrote:

> I wouldn't necessarily give up on a "record now" button, though it's  
> not necessarily feasible right now. I would have sworn there was  
> an upcoming standard for access to local microphones and webcams  
> due to hit at least Firefox in the relatively near future.

Could use a flash plugin (ducks)

> Though if there's a copy of ffmpeg executable by PHP on the server  
> I could have the script pre-report file duration and format and such...

All we need is the file size from the form. That way we know if it's  
uploaded or not.

> Okay, so I'll have some sort of (ironically enough) basic javascript-based  
> checker for javascript in the text on the submitter input form,  
> and a PHP-based, somewhat harsher checker on the "receiving script"  
> that will flag possibly-suspicious-looking uploads by prefixing the priority-level  
> with an "h" at the beginning of the directory name (right?).

>

Thinking about it lets try and avoid java script in general. I'd  
prefer the site to run fine completely without it.

Ken.