

CUNY Academic Commons - Feature #4340

embedding a video

2015-07-28 10:25 AM - Marilyn Weber

Status:	Resolved	Start date:	2015-07-28
Priority name:	Normal	Due date:	
Assignee:	Daniel Jones	% Done:	0%
Category name:	WordPress - Media	Estimated time:	0.00 hour
Target version:	1.8.14		
Description			
Ola Galal is logging in to CUNY Commons and trying to embed a video in the webpage - "The Center for place, culture, and politics". The video is taken from Videostreams.gc.cuny.edu. However, instead of the video, she just gets the link on the webpage. How does one embed the video itself instead of just link?			
Ola can be reached at 347-205-0850 or ola.galal@yahoo.com if you'd like to talk directly.			
Related issues:			
Has duplicate CUNY Academic Commons - Feature #4615: Plugin to allow for GC I...		Duplicate	2015-09-15

History

#1 - 2015-07-28 10:26 AM - Matt Gold

- Status changed from New to Assigned
- Assignee set to Boone Gorges

Hi Boone,

I seem to recall that you created a shortcode for such videos in the past

the correct URL is videostreaming.gc.cuny.edu

#2 - 2015-07-30 10:02 AM - Boone Gorges

- Category name set to WordPress - Media
- Status changed from Assigned to Reporter Feedback
- Target version set to Not tracked

Hi Marilyn - Can I get some more details on what the user is trying to do here? It appears to me that videos are in fact being embedded on the site in question. See eg <http://videostreaming.gc.cuny.edu/videos/video/3589/>. Perhaps the user could give specific instructions on how to reproduce the issue?

#3 - 2015-07-30 10:23 AM - Matt Gold

Hi Boone -- she wants to embed videos **from** videostreaming.gc onto her Commons site. The video streaming site itself is not on the Commons

#4 - 2015-07-30 10:32 AM - Boone Gorges

- Tracker changed from Support to Feature
- Status changed from Reporter Feedback to Assigned
- Assignee changed from Boone Gorges to Daniel Jones
- Target version changed from Not tracked to 1.9

Ah, thanks.

Dan, you want to look at this? There are two options here. One is to write a shortcode that mimics the iframe/embed behavior of the videos as they appear on eg <http://videostreaming.gc.cuny.edu/videos/video/3589/>. You can see some examples of our existing custom shortcodes at <https://github.com/cuny-academic-commons/cac/blob/1.8.x/wp-content/mu-plugins/assets/shortcodes.php>.

Alternatively, you could write something that apes WP's oEmbed logic. That is: when a videostreaming.gc.cuny.edu URL appears on its own line, swap it out with the iframe. It's not true oEmbed because **we** would be building the markup instead of requesting it from the oEmbed provider, but it would work the same way from the user's point of view. There's probably a way to tap into WP's native oEmbed detection regex so that you have to write a minimal amount of code to make it work. I've copied Ray here - he's done extensive work in this part of WP, and can probably point you in the

right direction.

This can be bumped back to a minor release when you're done with it.

#5 - 2015-08-04 12:42 PM - Raymond Hoh

There's probably a way to tap into WP's native oEmbed detection regex so that you have to write a minimal amount of code to make it work. I've copied Ray here - he's done extensive work in this part of WP, and can probably point you in the right direction.

Dan - the function you'll want to look at is `wp_embed_register_handler()`:
https://codex.wordpress.org/Function_Reference/wp_embed_register_handler

This adds embed support to WordPress by using regex for the URL matching and allows you to add a function callback to specify what you want to do with the URL for embedding.

Let me know if you have any questions about this.

#6 - 2015-08-08 06:52 PM - Daniel Jones

That is really nifty! Thanks so much Ray! I think I've got it working here:
<https://github.com/cuny-academic-commons/cac/commit/0035abc7cec43aeb3730333273678c4df231aa3e>

I put it in `functions.php`, but do we want it in its own plugin?

#7 - 2015-08-08 07:04 PM - Daniel Jones

Wait of course we want it in its own plugin. My bad - put it in `mu-plugins`:
<https://github.com/cuny-academic-commons/cac/commit/e96c052bd7f33edd115364eb5f58450bcf3817f6>

#8 - 2015-08-11 02:52 PM - Boone Gorges

- *File videostreaming-error.jpg added*

Thanks, Dan! Looking good so far. But for me, the video embed is failing. I've attached a screenshot of the error message I see.

I'm guessing that this error is being thrown because the `access_token` is time-sensitive. When you first grabbed the `iframe` markup, you had a fresh token, so embedding worked in your tests. But now it's expired, so it's broken for me :)

Access tokens like this are explicitly designed so that people don't hotlink content, like we're attempting to do here. Maybe we should reach out to them to ask if we can remove the token requirement for requests coming from the Commons. Matt, what do you think?

Alternatively, we could scrape the source page to get a fresh access token. It's a bit icky to do this when it appears to be explicitly against their wishes, but it's definitely an option if the folks who host the videos are OK with it (but don't want to modify their software to allow native hotlinking for the Commons).

#9 - 2015-08-11 03:32 PM - Matt Gold

If I need to make a request of A/V at the GC about this, please just let me know what I need to ask. Thanks.

#10 - 2015-08-11 04:03 PM - Boone Gorges

Thanks, Matt. You could point them to this ticket, with the following summary:

A member of the CUNY Academic Commons has requested the ability to embed a video hosted on `videostreaming.gc.cuny.edu` directly on her `commons.gc.cuny.edu` site. Our dev team tried building a tool that would embed the video, but streaming was blocked due to an outdated `access_token`. We assume that the `access_token` is there to prevent hotlinking, but we're hopeful that you can make an exception in the case of the Commons. Is it possible to whitelist embeds from the Commons's IP address (146.96.128.200)? If, for technical reasons, you're not able to create a whitelist, it's possible for our team to scrape your pages in order to harvest a valid `access_token`. But we don't want to take this step without your blessing.

Please modify as you see fit, and feel free to put them directly in touch with me, either over email or through this ticket.

#11 - 2015-08-11 04:12 PM - Matt Gold

Email sent. Thanks, all.

#12 - 2015-08-17 02:57 PM - Daniel Jones

Thanks, Boone and Matt! Sorry, should have realized that the access token would expire. Wishful thinking on my part ;-). Happy to pick this back up when you hear back, Matt!

#13 - 2015-08-31 09:56 AM - Marilyn Weber

Matt -
Did you ever hear back about this from GC A/V? Ola has contacted me again. Thanks!
- Marilyn

#14 - 2015-08-31 10:18 AM - Boone Gorges

Hi Marilyn - I had brief contact with Joe from GC IT, but we kinda lost the thread. I'll follow up.

In the meantime, please let Ola know that we're looking into this, but that we can't make any promises about when it'll be implemented. Until it is, she'll have to use links to the videostreaming page instead of embeds.

#15 - 2015-08-31 01:20 PM - Matt Gold

I received a phone call from A/V and need to get back to them

#16 - 2015-09-08 09:08 AM - Matt Gold

Hi Boone,

I think you were back in touch with A/V. Was any resolution reached?

#17 - 2015-09-08 02:50 PM - Boone Gorges

I spoke briefly with Joe. After running through a bit of troubleshooting, he passed my request along to his tech team. I haven't heard anything yet.

#18 - 2015-10-01 11:16 AM - Boone Gorges

- *File API Documentation.docx added*

API documentation attached. More to come in a moment.

#19 - 2015-10-01 11:35 AM - Boone Gorges

OK, I think we're ready to roll. Dan, here's where we stand.

I've spoken with the fine folks in IT who manage the videostreaming site. It turns out that the application they use has a public-facing API (one that IT didn't know they had!). We've been granted a specific set of credentials for accessing the API. You'll be using the credentials to fetch an `access_token`, which can then be used to generate embeds in the way you're already doing.

The API offers lots of stuff (see the docs I uploaded a few minutes ago) but we're only going to be using the initial OAuth2 handshake, using `grant_type=password`, in order to fetch an `access_token`. The request URL looks like this:

```
https://videostreaming.gc.cuny.edu/api/v1/?grant_password=password&username=$cac_username&password=$cac_password
```

where `$cac_username` and `$cac_password` are our credentials. I'm going to store the credentials as constants in our `wp-config.php` file, so you can access them using `CAC_VIDEOSTREAMING_USERNAME` and `CAC_VIDEOSTREAMING_PASSWORD`. Use `add_query_arg()` to build the URL, and use `wp_remote_get()` to make the request. Make sure to use the HTTPS URL - OAuth2 depends on SSL.

The response body (`wp_remote_retrieve_body()`) will contain the `access_token` you need, plus an expiry in seconds. I recommend that you use the following logic when building your markup:

```
$token_data = get_site_option( 'cac_videostreaming_token_data' );

// Assume we don't have a valid token.
$token_is_valid = false;
if ( isset( $token_data['expires'] ) && time() < $token_data['expires'] ) {
    // The token is OK if the expiry is in the future.
    $token_is_valid = true;
    $token = $token_data['token'];
}

if ( ! $token_is_valid ) {
    // Build the URL and make the request
    $request_url = whatever;
    $request = wp_remote_get( $request_url );
    // maybe check that the response code is 200, then get the token and expiry from the body
    $token = the access token sent back from the API

    $token_data = array(
        'token' => $token,
        'expires' => whatever // I think you should add the expiration in seconds to the timestamp of the response. Round down if in doubt :)
    );
}
```

```
);  
update_site_option( 'cac_videostreaming_token_data', $token_data );  
}
```

// Now you have a valid \$token and can build your markup

This logic will minimize authentication requests on their API - they should only happen once a day at most (token expiry is 24hrs).

Thanks, Dan!

#20 - 2015-10-01 03:54 PM - Boone Gorges

Sorry, forgot to add the creds before. They're now in wp-config

<https://github.com/cuny-academic-commons/cac/commit/c518b4ed35acbf4c237d9363bbb115c6d9b9b644>

#21 - 2015-10-19 06:30 PM - Daniel Jones

Okay thanks so much Boone for all the help with this! I think I've got it working here:

<https://github.com/cuny-academic-commons/cac/commit/d9f6bc78a1084e57f6f75e8bc5c99f31bd434913>

Let me know if you see any issues! And thanks again!

#22 - 2015-10-20 12:01 AM - Boone Gorges

- Status changed from Assigned to Resolved

- Target version changed from 1.9 to 1.8.14

Thanks, Dan!

This looks mostly good. I had a slight problem getting embeds to actually work, due to what I think is a bug/config error in the videostreaming.gc.cuny.edu application. The embed_code that the API was generating had URLs that looked like 'http://172.29.28.54...'. This URL format doesn't work: the videostreaming.gc.cuny.edu API expects requests to occur over HTTPS, and HTTPS requests can only take place against the canonical domain, videostreaming.gc.cuny.edu. I added some ugly logic in

<https://github.com/cuny-academic-commons/cac/commit/2bd7a29cb2ec66829924826d56848f5c3b2e11d1> that makes the necessary swaps. I will drop an email to the administrator for the API to let them know that their config probably needs to be updated, so that the embed_code returned is valid.

I think this is looking good, so I'm going to move it to a 1.8.x release. It was too complicated to cherry-pick, so I plopped it over wholesale :)

<https://github.com/cuny-academic-commons/cac/commit/9b00a50cd19c69592ba4b394d463e27d9930b0cf>

Thanks for your work on this issue!

Files

videostreaming-error.jpg	35.2 KB	2015-08-11	Boone Gorges
API Documentation.docx	25.8 KB	2015-10-01	Boone Gorges