

Pic Puller for Craft 3 Documentation

Overview

Pic Puller for Craft 3 is a plug in for Craft CMS that lets authorized users pull in their Instagram media into Craft.

Pic Puller provides native Craft tags to access Instagram media in templates. It also includes a field type that allows authorized users to browse their personal media stream on Instagram.

What's new in version 3

Pic Puller for Craft 3 has been rewritten for Craft CMS 3. Functionalty is the same as in version 2.

For the sake of history, there was a significantly different version of Pic Puller in version 1 but ended due to Instagram's changed terms of API use. In late 2015, Instagram scaled back access to their API. Instagram now requires approval of all apps that attempt to use their API. Instagram would not approve any versions of Pic Puller 1 apps that users had created.

After scaling back app permissions, Pic Puller's API access was approved by Instagram.

All instances of the Pic Puller plugin now use this new Instagram-approved API access.

Updating from version 2.x

Your Twig template code should be about the same. The one change you need to make is relatively minor.

The plugin handle is changing from `picpuller` to `picPuller`.

So, if you had an opeing Pic Puller tag in Craft 2 like this:

```
{% for instagramdata in craft.picpuller.media_recent({'user_id' : 1,
'use_stale_cache' : true, 'limit': 4, 'max_id': nextmaxid }) %}
```

It would now be like this, with an uppercase “P” in the handle name:

```
{% for instagramdata in craft.picPuller.media_recent({'user_id' : 1,
'use_stale_cache' : true, 'limit': 4, 'max_id': nextmaxid }) %}
```

Limitations

The API access that Pic Puller has access to is basically limited to an authorized user’s media. There is no access available to other user’s public content. *Instagram does not permit searching via #hashtag for public media through Pic Puller.*

There is no “hashtag” searching available, even for tags in your own media.

Getting help

If you run into any problems using Pic Puller, please drop a line at john@johnfmorton.com and mention “Pic Puller for Craft support” in the subject line and I’ll get back to you ASAP.

I’m always interested in how people use Pic Puller. If you send your URL to me, I’d love to see it.

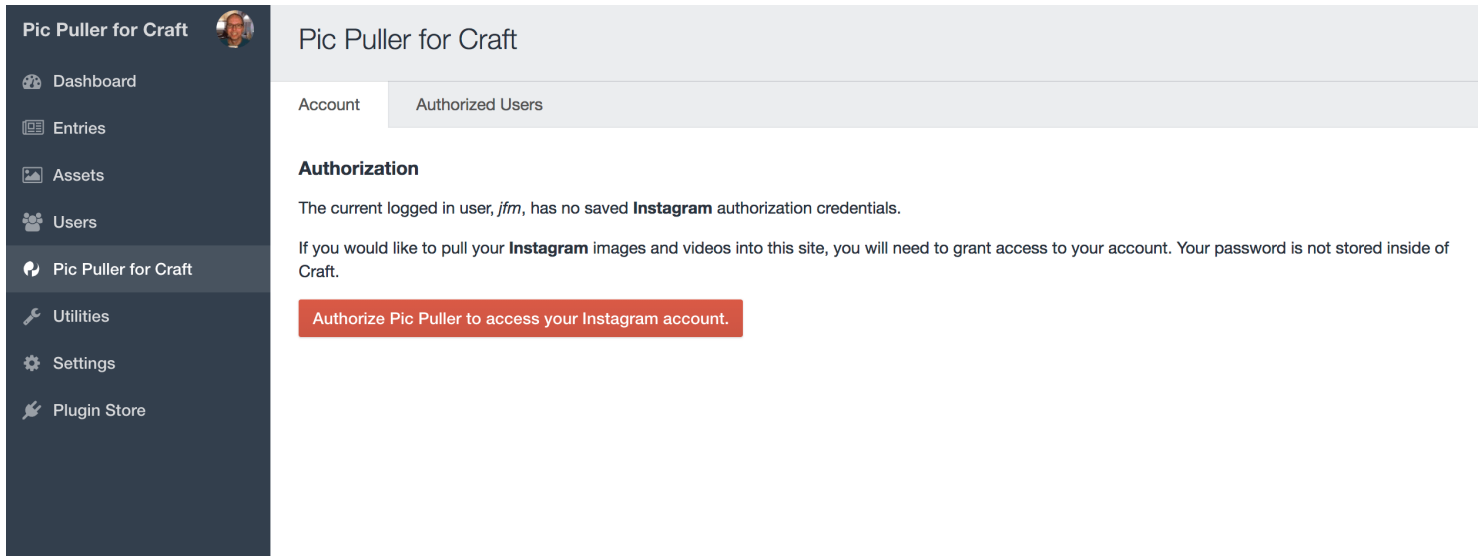
Thanks - John

Shameless plug

If you don’t already subscribe to **Craft Link List**, please check it out at <http://craftlinklist.com>. It’s free and will keep you up-to-date on the latest links from the Craft community.

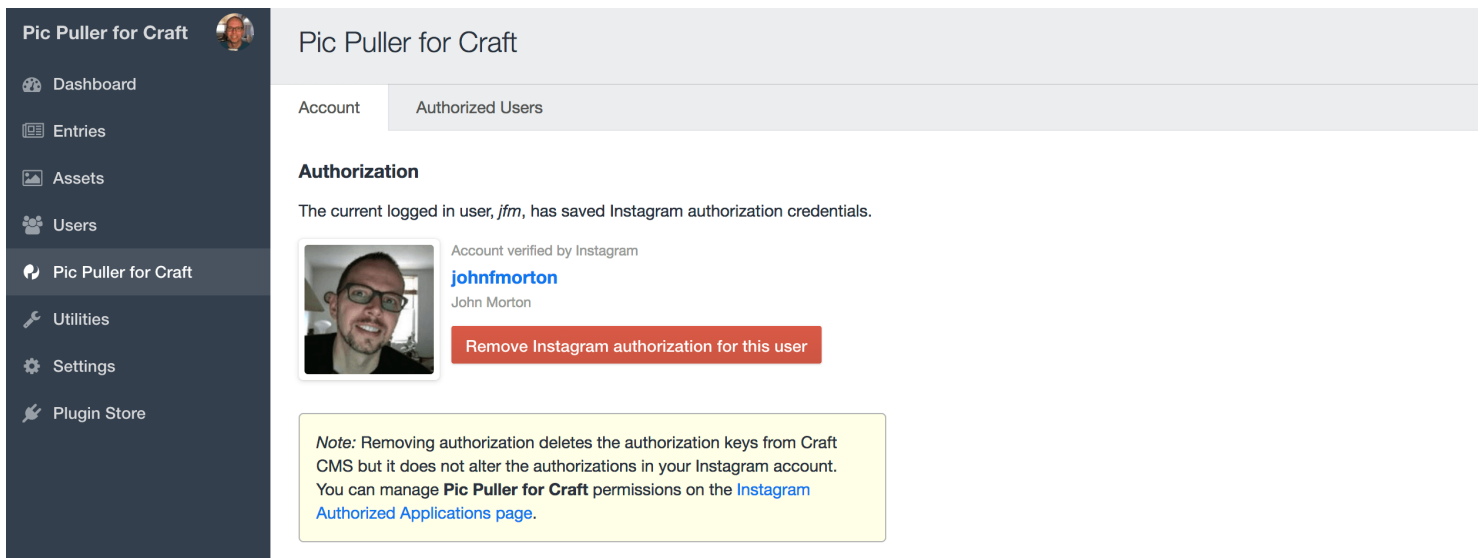
Authorizing with Instagram

To authorize a Craft site to pull in Instagram media, a user must allow access. After installing Pic Puller, the site will ask for authorization.



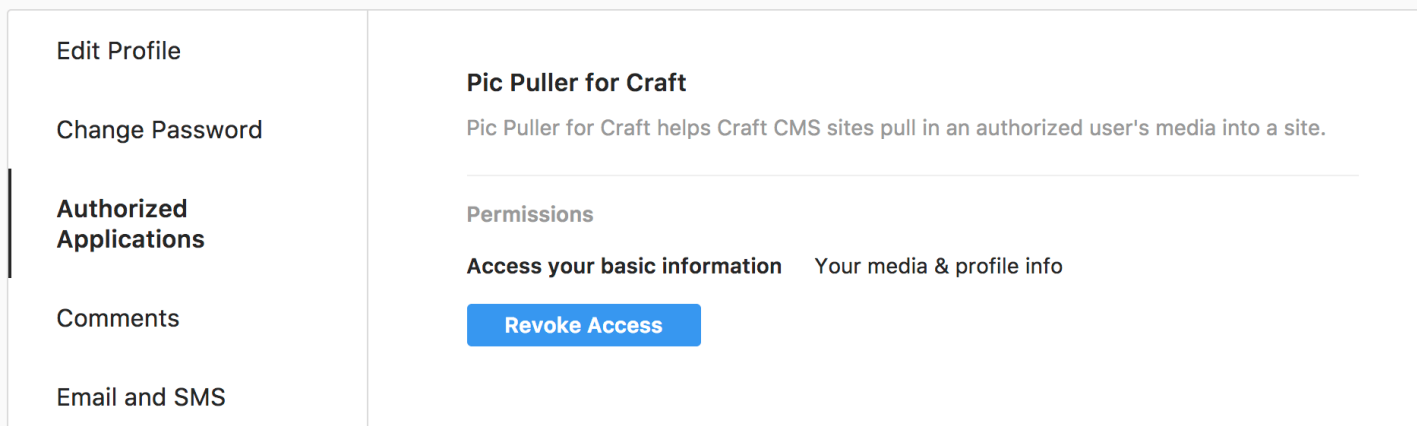
Pic Puller for Craft “Account” tab in the Craft control panel

Once authorization has been granted, Pic Puller will display the authorized user’s credentials.



Pic Puller for Craft “Account” tab in the Craft control panel

Users will see *Pic Puller* as an authorized application in their Instagram account.



Pic Puller for Craft in the Instagram Authorized Application page

Multiple users and Pic Puller for Craft

You are not limited to one user with Pic Puller. If you have purchased the “Craft Client” or “Craft Pro” package, i.e. the paid version of Craft, additional users can also authorize their Instagram accounts. Pic Puller works without a the paid Craft CMS version but is limited to the one user allowed by that version of the Craft software.

A multiuser site can have each user display their own personal media on the site. When using the Pic Puller Image Browser the user stream search will be from each authorized user’s Instagram stream.

Any “Admin” level user can view the list of authorized Pic Puller users along with their Instagram OAuth key for the Pic Puller for Craft application. Basically, you need to trust other Admins in your Craft site.

Users that have not been granted “Admin” rights within Craft can still access the Pic Puller plugin page as long as you have granted them access under the “Users” tab. Access can be granted on a one-by-one basis under the “Users” tab, or on a broader scale if you set up User Groups and provide the group access to Pic Puller.

These non-admin users will have the ability to authorize and de-authorize Pic Puller with an Instagram account. Non-admin users will not be able to see the list of authorized users across the entire Craft site.

The screenshot shows the 'Pic Puller for Craft' sidebar on the left with the 'Authorized Users' tab selected. The main content area is titled 'Authorized users' and states: 'There are 2 users who have authorized this site for access to their Instagram accounts.' Below this is a table with the following data:

Craft User ID	Craft User	Instagram ID	Instagram oAuth
☆ 1	John Morton (<i>jfm</i>)	1200012	1200012. 1200012.abcdcd9[redacted]
5	Jane Joe (<i>janedoe</i>)	123456710	123456710.abcd123.efga5fh[redacted]

Pic Puller for Craft “Authorized Users” tab in the Craft control panel

Optional behavior for multiple users

The screenshot shows the 'Pic Puller for Craft' settings panel. The 'Sharing options' section contains the following settings:

- Short name for Pic Puller for Craft.**
Enter an alternate short name for the Pic Puller plugin, like "PPfC" or "Instagram".
Input field: Pic Puller for Craft
- Let users share a single Instagram Authorization? (default: off)**
The normal behavior of Pic Puller is to have each Craft user authorize with their own Instagram account. You may override this to share a single Instagram account across the entire Craft site.
Toggle switch: ☒ (indicated by a red arrow)
- Select account for sharing oAuth. (Setting is ignored if setting above is off).**
If sharing a single oAuth, select an Admin user to serve as the user whose oAuth credentials will be used across the site for all users.
Radio buttons: ☒ jfm (John Morton) and ☐ Jane Doe (janedoe)

As described above, the default behavior of Pic Puller for Craft is for each Craft user to authenticate their Craft account with their Instagram account.

You can override this default behavior and have a single oAuth authorization be shared amongst all users of the Craft site. The most likely scenario for this feature is of a company site that has many site editors, but only a single Instagram account across the organization. Choosing to use a single Instagram authorization requires that the site developer choose a single user as the master Instagram account holder on the site. **This user must be an Admin.** The selection of the master Instagram user is made from the Pic Puller settings panel.

Only the selected account holder will see the “Pic Puller for Craft” menu item in the global navigation. This is the menu that allows a user to create and manage the Instagram application and authorization.

Tip: If your Instagram account holder in your organization is not someone who would typically have an admin level account to the site, one suggestion is to create an admin level account to be used solely for authorization purposes. You can have the Instagram account holder log into this account *once* to have them authorize Pic Puller to access the corporate Instagram account. Under everyday circumstances you could have them use their regular “editor” account to contribute to the site.

Pic Puller Image Browser Field type

The Pic Puller Image Browser field type lets an authorized user browse their Instagram media stream and select media for use in a Craft entry.

When setting up a field select “Pic Puller: Instagram Browser” from the Field Type dropdown.

Pic Puller for Craft

Dashboard

Entries

Assets

Users

Pic Puller for Craft

Utilities

Settings

Plugin Store

Settings > Fields > Pic Puller Tests >

Instagram Browser

Save

Group

Which group should this field be displayed in?

My Fieldtype Group

Name

What this field will be called in the CP.

Instagram Browser

Handle

How you'll refer to this field in the templates.

instagramBrowser

Instructions

Helper text to guide the author.

Choose an image from Instagram

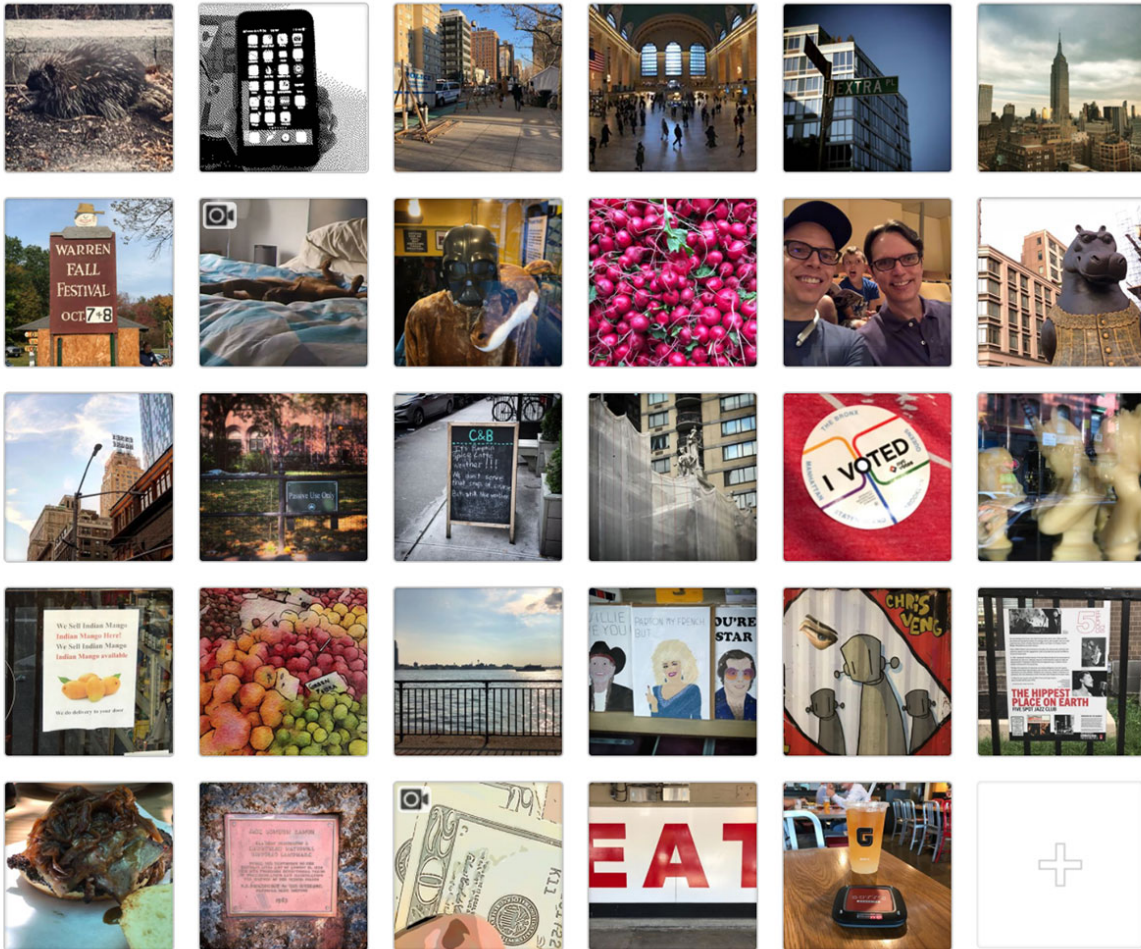
Field Type

What type of field is this?

Pic Puller: Instagram Browser

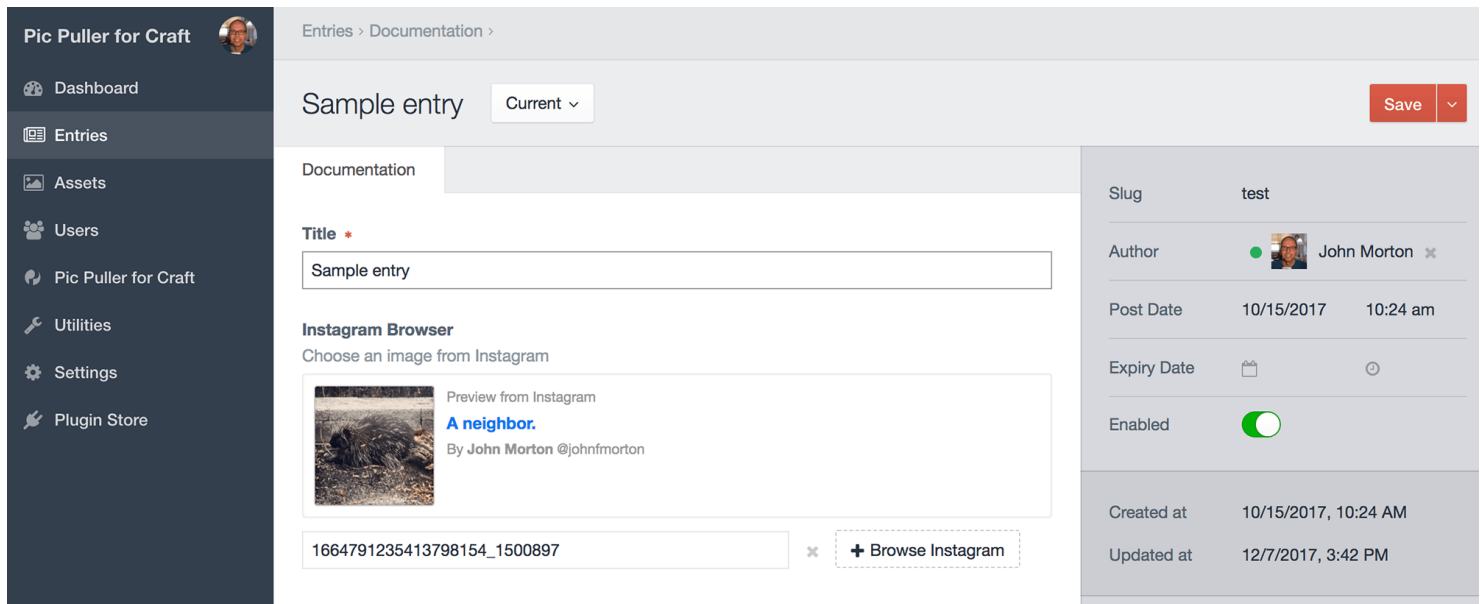
Creating a Pic Puller Image Browser field type

Instagram Images



Cancel

The Pic Puller Instagram Browser



The Pic Puller Instagram Browser Thumbnail Preview

The field type searches Instagram to retrieve a media ID for an image or video uploaded through Instagram. A small video icon in the upper lefthand corner designates that the piece of media is a video rather than an image.

The media ID is stored in the Craft database. The media file is not stored within your site as dictated by the Instagram API agreement.

The media ID is used in conjunction with the media by ID function in your templates to show media on your site. (See the **Media by ID** section of the documentation below.)

Working with Pic Puller for Craft in your templates

Caching of Pic Puller requests & Instagram rate limits

Although Pic Puller does not store Instagram media, it does cache the JSON data that is returned from successful Instagram API requests.

Instagram limits the number of requests that can be made to it's API endpoints. See <https://www.instagram.com/developer/limits/> for the full details. These limits reset every hour, but exhausting the allotted requests could result in your site not having the Instagram media you expect.

If you use the `use_stale_cache` parameter set to `true` in your Pic Puller tags, the cached JSON data will be used when Instagram does not deliver valid data back from a request.

Pic Pullers's JSON caching does not replace Craft's own [caching tags](#). You should also use Craft's own caching in your templates.

Template functions

There are 3 template functions available with Pic Puller.

User information

craft.picPuller.user

Description: Get basic information about a user.

Instagram docs page for this function:

https://www.instagram.com/developer/endpoints/users/#get_users

Required parameters

user_id: the Craft user id (*not* an Instagram user id)

Optional parameters

use_stale_cache: BOOLEAN, either TRUE or FALSE (defaults to TRUE if undefined), to have Pic Puller use previously cached data returned in the event of an error in retrieving new data

Tags returned in a successful Craft loop:

status: a BOOLEAN of TRUE (1) is returned when Instagram media data is returned, *even if it is cached data*

username: the Instagram username

id: the Instagram user id

bio: biography information provided by the Instagram user

profile_picture: URL to the profile image of the user

full_name: the full name provided by the user on Instagram

cacheddata: a BOOLEAN of TRUE (1) is returned when request is using cached data.

counts_media: the total number of images in this user's Instagram feed

counts_followed_by: the number of users who follow this user on Instagram

counts_follows: the number of users this user follows on Instagram

website: the website URL provided by the user whose account the image originates from

error_type: a string of “NoError” to indicate a successful call to the Instagram API resulting in valid data

error_message: a string describing the *lack* of an error being returned

Tags returned in an unsuccessful Craft loop:

status: a BOOLEAN of FALSE (0) is returned when no data is returned from Instagram or there is no cache data to return

error_type: a single code word indicating the type of error (“MissingReqParameter”, “UnauthorizedUser”, “NoCodeReturned” issued by Pic Puller. Other error codes are passed through from Instagram.)

error_message: a string describing the error

Example template code:

```
{% for instagramdata in craft.picPuller.user({'use_stale_cache' : true, 'user_id' : 1
}) %}
  <p>status: {{ instagramdata.status }}</p>
  <p>cacheddata: {{ instagramdata.cacheddata }}</p>
  {% if instagramdata.status == TRUE %}
    <p>username: {{ instagramdata.username }}</p>
    <p>full_name: {{ instagramdata.full_name }}</p>
    <p>profile_picture:</p>
    <p></p>
    <p>counts_media: {{ instagramdata.counts_media }}</p>
  {% else %}
    <p>Error Type: {{ instagramdata.error_type }}</p>
    <p>Error Message: {{ instagramdata.error_message }}</p>
  {% endif %}
{% endfor %}
```

Recent media

craft.picPuller.media_recent

Description: Get the most recent media published by a user.

Instagram docs page for this function:

https://www.instagram.com/developer/endpoints/users/#get_users_media_recent

Required parameters

user_id: the Craft user id (*not* an Instagram user id)

Optional parameters

limit: an integer indicating how many images to request from Instagram. Instagram may return fewer under some circumstances (See *Unexpected number of images returned* in the FAQ at the of this document). Maximum of 32 allowed by Instagram.

use_stale_cache: BOOLEAN, either TRUE or FALSE (defaults to TRUE if undefined), to have Pic Puller use previously cached data returned in the event of an error in retrieving new data

max_id: an integer used to determine pagination of results. (See next_max_id in the 'Tags returned' below section for more information.)

Tags returned in a successful Craft loop:

status: a BOOLEAN of TRUE (1) is returned when Instagram media data is returned, even if it is cached data

type: returns a string "image" or "video"

media_id: the Instagram unique media ID for the image or video

created_time: time stamp of image creation time, Unix timestamp formatted

link: URL of the media's page on Instagram

cacheddata: a BOOLEAN of TRUE (1) is returned when request is using cached data.

caption: the caption provided by the author. Note, it may be left untitled which will return an empty string.

caption_only: the caption provided by the author *minus* any content starting with the first hashtag. For example, if the caption were "Digging into Craft CMS. #craftcms #code #fun", the caption_only function would return "Digging into Craft CMS."

tags: an array of the tags associated with the media

thumbnail: URL to image

thumbnail_width: width of image in pixels.

thumbnail_height: height of image in pixels.

low_resolution: URL to image

low_resolution_width: width of image in pixels.

low_resolution_height: height of image in pixels.

standard_resolution: URL to image

standard_resolution_width: width of image in pixels.

standard_resolution_height: height of image in pixels.

video_low_bandwidth: URL to video

video_low_bandwidth_width: width of video in pixels.

video_low_bandwidth_height: height of video in pixels.

video_low_resolution: URL to video

video_low_resolution_width: width of video in pixels.

video_low_resolution_height: height of video in pixels.

video_standard_resolution: URL to video

video_standard_resolution_width: width of video in pixels.

video_standard_resolution_height: height of video in pixels.

latitude: latitude data, if available

longitude: longitude data, if available

next_max_id: an integer, provided by Instagram, used to return the next set in the same series of images. Pass this value into the max_id parameter of the loop to get the next page of results.

error_type: a string of "NoError" to indicate a successful call to the Instagram API resulting in valid data

error_message: a string describing the *lack* of an error being returned

Tags returned in an unsuccessful Craft loop:

status: a BOOLEAN of FALSE (0) is returned when no data is returned from Instagram or there is no cache data to return

error_type: a single code word indicating the type of error (“MissingReqParameter”, “UnauthorizedUser”, “NoCodeReturned” issued by Pic Puller. Other error codes are passed through from Instagram.)

error_message: a string describing the error

Example template code:

```
{% for instagramdata in craft.picPuller.media_recent({'user_id' : 1, 'use_stale_cache' : true, 'limit': 20}) %}
    {% if loop.first %}
        <p>Status: {{ instagramdata.status }}</p>
        <p>Error Type: {{ instagramdata.error_type }}</p>
        <p>Error Message: {{ instagramdata.error_message }}</p>
        <hr>
    {% endif %}
    {% if instagramdata.status == 'true' %}
        <p>Loop Index: {{ loop.index }}</p>
        {% if instagramdata.video_low_resolution != '' %}
            <p>This is a video; not an image.</p>
            <p>{{ instagramdata.video_low_resolution }}</p>
        {% endif %}
        <p></p>
        <p>caption: {{instagramdata.caption}}</p>
        <p>created_time: {{ instagramdata.created_time }}</p>
    {% else %}
        <p>Error Type: {{ instagramdata.error_type }}</p>
        <p>Error Message: {{ instagramdata.error_message }}</p>
    {% endif %}
{% endfor %}
```

Media by ID

craft.picPuller.media

Description: Get information about a single media object.

Instagram docs page for this function:

https://www.instagram.com/developer/endpoints/media/#get_media

Required parameters

user_id: This is the ID number of an Craft user. (It is not the Instagram user id number.)

media_id: this is the ID number that Instagram has assigned to an image or video

Optional parameters

use_stale_cache: BOOLEAN, either TRUE or FALSE (defaults to TRUE if undefined), to have Pic Puller use previously cached data returned in the event of an error in retrieving new data

Tags returned in a successful Craft loop:

status: a BOOLEAN of TRUE (1) is returned when Instagram media data is returned, even if it is cached data

created_time: time stamp of image creation time, Unix timestamp formatted

link: URL of the image's homepage on Instagram

cacheddata: a BOOLEAN of TRUE (1) is returned when request is using cached data.

caption: The caption provided by the author. Note, it may be left untitled which will return an empty string.

caption_only: the caption provided by the author *minus* any content starting with the first hashtag. For example, if the caption were "Digging into Craft CMS. #craftcms #code #fun", the caption_only function would return "Digging into Craft CMS."

tags: an array of the tags associated with the media_id

thumbnail: URL to image

thumbnail_width: width of image in pixels.

thumbnail_height: height of image in pixels.

low_resolution: URL to image

low_resolution_width: width of image in pixels.

low_resolution_height: height of image in pixels.

standard_resolution: URL to image

standard_resolution_width: width of image in pixels.

standard_resolution_height: height of image in pixels.

video_low_bandwith: URL to video

video_low_bandwith_width: width of video in pixels.

video_low_bandwith_height: height of video in pixels.

video_low_resolution: URL to video

video_low_resolution_width: width of video in pixels.

video_low_resolution_height: height of video in pixels.

video_standard_resolution: URL to video

video_standard_resolution_width: width of video in pixels.

video_standard_resolution_height: height of video in pixels.

latitude: latitude data, if available

longitude: longitude data, if available

username: the Instagram username of the user whose account the image originates from

user_id: the Instagram user id of the user whose account the image originates from

full_name: the full name provided by the user whose account the image originates from

profile_picture: URL to the profile image of the user

likes: number of likes for piece of media

error_type: a string of "NoError" to indicate a successful call to the Instagram API resulting in valid data

error_message: a string describing the error

Tags returned in an unsuccessful Craft loop:

status: a BOOLEAN of FALSE (0) is returned when no data is returned from Instagram or there is no cache data to return

error_type: a single code word indicating the type of error (“MissingReqParameter”, “UnauthorizedUser”, “NoCodeReturned” issued by Pic Puller. Other error codes are passed through from Instagram.)

error_message: a string describing the error

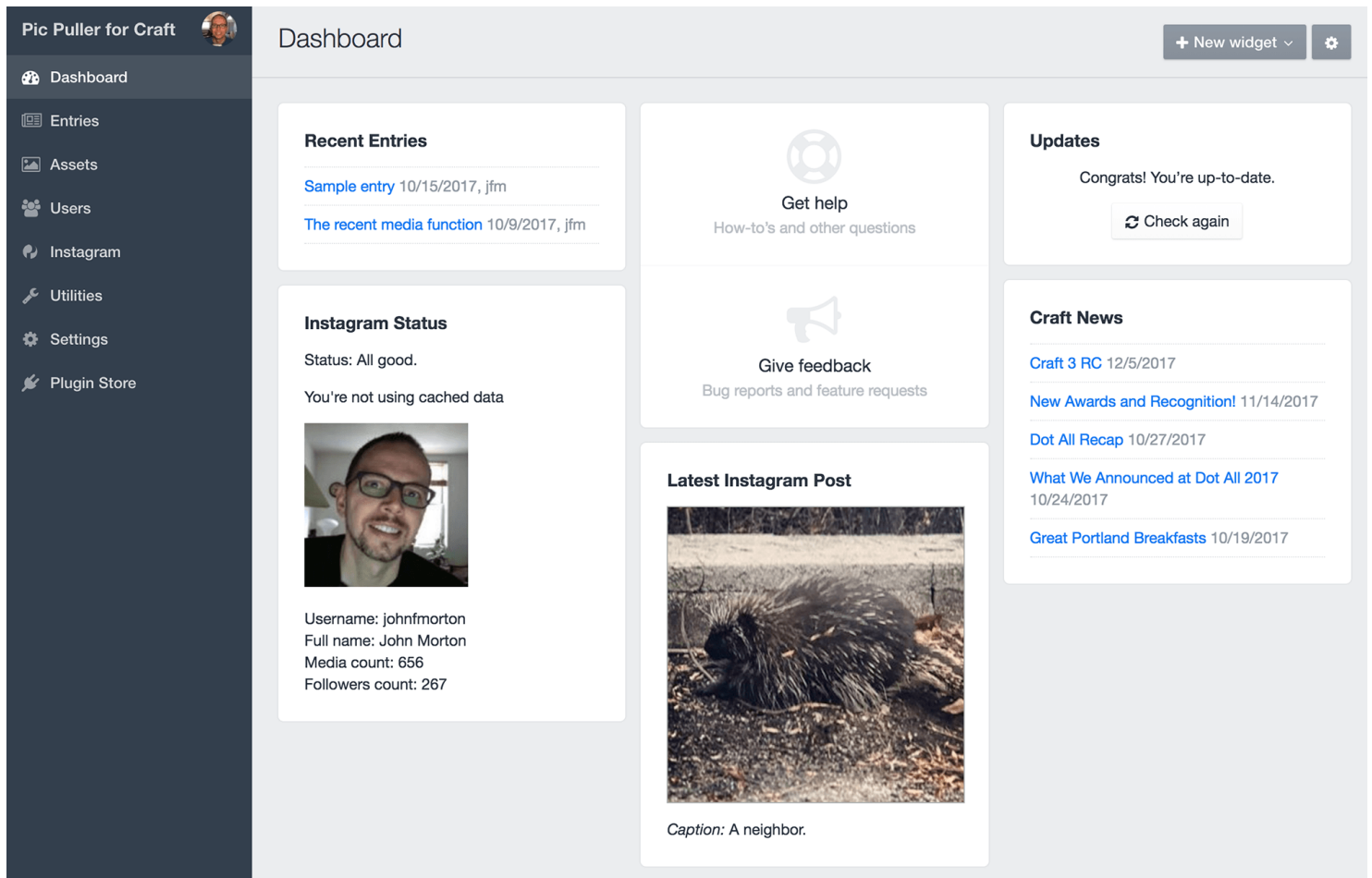
Example template code:

```
{% for instagramdata in craft.picPuller.media({'use_stale_cache' : TRUE, 'user_id' : 1
, 'media_id': '423894109381331599_1500897'}) %}
    {% if loop.first %}
        <p>Status: {{ instagramdata.status }}</p>
        <p>Error Type: {{ instagramdata.error_type }}</p>
        <p>Error Message: {{ instagramdata.error_message }}</p>
        <hr>
    {% endif %}
    {% if instagramdata.status == 'true' %}
        <p>username: {{ instagramdata.username }}</p>
        <p>full_name: {{ instagramdata.full_name }}</p>
        <p></p>
        <p>caption: {{instagramdata.caption}}</p>
        <p>created_time: {{ instagramdata.created_time }}</p>
    {% else %}
        <p>Error Type: {{ instagramdata.error_type }}</p>
        <p>Error Message: {{ instagramdata.error_message }}</p>
    {% endif %}
{% endfor %}
```

Pic Puller Options

Control Panel Widgets

Pic Puller includes 2 control panel widgets. The Latest Image widget that displays the most recent image from an authorized user’s Instagram feed. It is added on a per user basis to the Dashboard home screen in the control panel. The Instagram Status widget displays an authorized Instagram user’s user data.



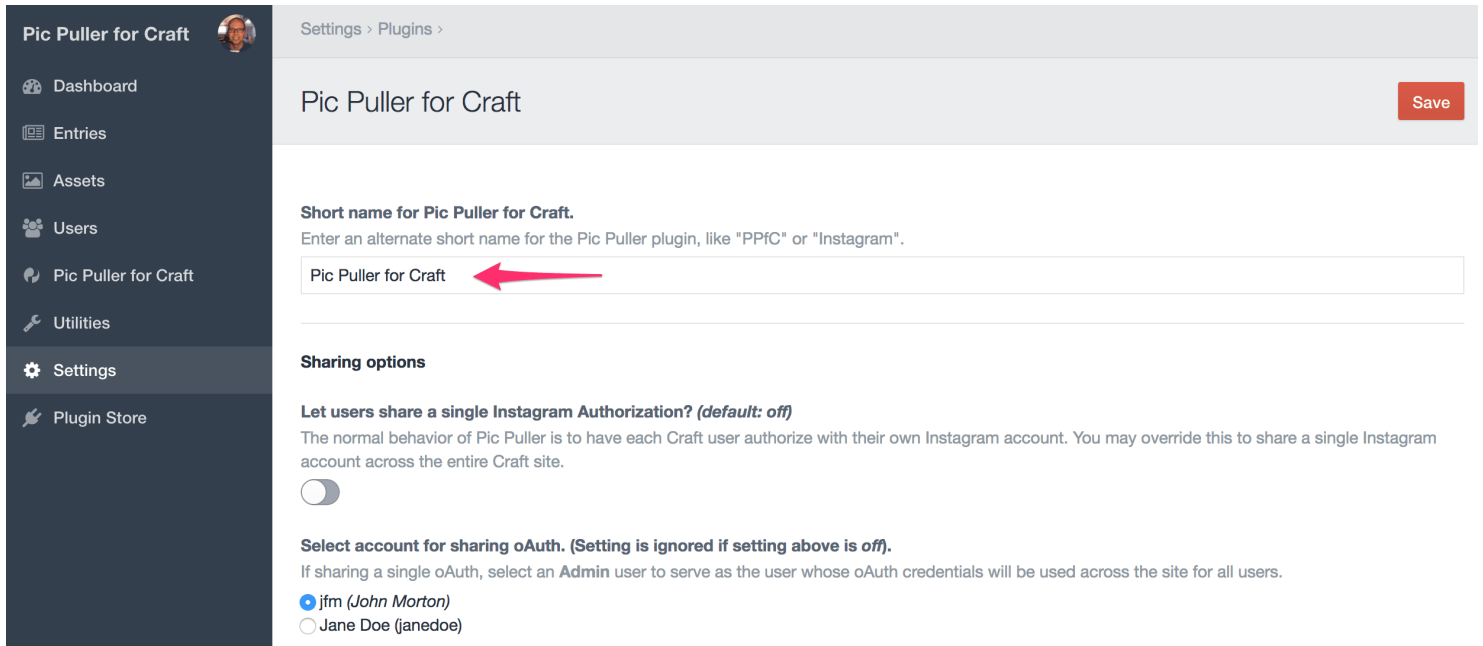
The Pic Puller widget

Customize the plug in name

You can easily customize the name “Pic Puller for Craft” to be shorter. You’ll need to be an Admin to do this. It will affect all users.

Click the gear icon and go to the Plugins page within the control panel.

Now click the name of the plugin, Pic Puller for Craft.



Settings > Plugins >

Pic Puller for Craft

Short name for Pic Puller for Craft.
Enter an alternate short name for the Pic Puller plugin, like "PPfC" or "Instagram".

Pic Puller for Craft

Sharing options

Let users share a single Instagram Authorization? (default: off)
The normal behavior of Pic Puller is to have each Craft user authorize with their own Instagram account. You may override this to share a single Instagram account across the entire Craft site.

☐

Select account for sharing OAuth. (Setting is ignored if setting above is off).
If sharing a single OAuth, select an **Admin** user to serve as the user whose OAuth credentials will be used across the site for all users.

☒ jfm (John Morton)
☐ Jane Doe (janedoe)

Create a new name for Pic Puller

Change it to whatever you like “PPfC”, Pic Puller“ or even just ”Instagram“. Then click Save. You will see the change immediately. If you leave the name blank, it will default back to the full name of the plugin.

FAQs

Unexpected number of images returned

The number of images returned from a Pic Puller loop can be up to 32 images in most cases, but it’s not guaranteed. Why is that? It has to do with how the Instagram API returns data. Although I’ve not read it in official documentation, it appears whatever number of images you request are returned from the API and then that set of images is filtered for images that may have been deleted by users. For example, if I were pulling 3 images from my Instagram feed, I would expect 3 images returned in my Pic Puller loop in Craft. If I went into my Instagram app on my phone and deleted the most recent photo I’ve taken, I would then only receive 2 images back in my Craft site.

When will tag searching return to Pic Puller?

I’m doubtful that Instagram will re-add the hashtag search to this version of the API.