
Plone Developer Manual Documentation

Release

Plone community and individual contributors

December 13, 2012

CONTENTS

This document is intended for Plone developers who want to:

- let Plone contents be OpenGraph enabled
- let Facebook user be authenticated in Plone

`collective.fb` is a bundle that:

1. includes the `collective.opengraph` package
2. will include another PAS plugin package that will let users login using their Facebook account

The `collective.opengraph` package will also be a dependency for the PAS plugin.

THE TWO SIDES

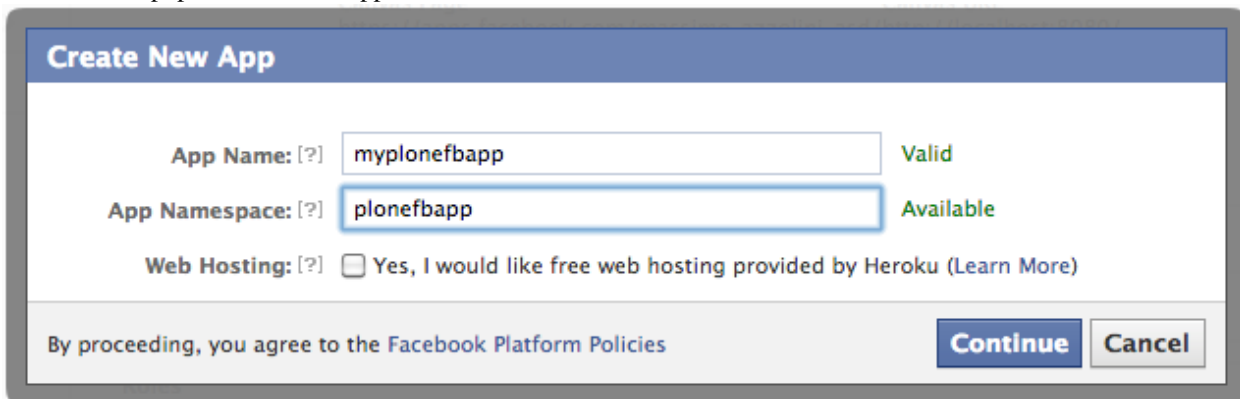
Of course, you have to proceed into two steps on two different platform.

1.1 The Facebook side

1.1.1 How to configure Facebook

The very first thing you have to do is to create a Facebook application.

1. Open <https://developers.facebook.com/app> (if you are not registered yet, do it [here](#)). Look at that 'Create new app' button. It's your friend.
2. Fill the popover form that appears



Create New App

App Name: [?] myplonefbapp Valid

App Namespace: [?] plonefbapp Available

Web Hosting: [?] Yes, I would like free web hosting provided by Heroku (Learn More)

By proceeding, you agree to the Facebook Platform Policies **Continue** Cancel

3. Pass the security check (captcha). This is what you get:

Apps ▶ myplonefbapp [Edit App](#) [+ Create New App](#)

Settings [Edit Settings](#)


Summary	App ID/API Key	App Secret
	App Namespace	Contact Email
	plonefbapp	massimo.azzolini@gmail.com
	Support Email	App Description
	massimo.azzolini@gmail.com	

Open Graph [Edit Open Graph](#)

You have not added any actions, objects, or profile units. [Get started](#) using the Open Graph.

Roles [Edit Roles](#)

Roles Administrators:



Insights [See All](#)

Users	Sharing
1 Daily New Users	0 Daily Content Shared
0 Daily Active Users	0.00 Feedback per Share

4. Edit the settings and complete the basic configuration with few more data:
- the domain of your website (e.g. mywebsite.com)
 - (opt.) if you want to have Facebook authentication, select “Website with Facebook Login” and set the URL of your website.
 - (opt.) choose a category

Object Debugger

Home > Tools > Debugger

Input URL, Access Token, or Open Graph Action ID

Debug


Scrape Information

Response Code: 200
 Fetched URL: <http://www.facciamoadesso.it/notizie/e-online-la-guida-per-le-imprese-colpite-dal-terremoto>
 Canonical URL: <http://www.facciamoadesso.it/notizie/e-online-la-guida-per-le-imprese-colpite-dal-terremoto>

Open Graph Warnings That Should Be Fixed

Small og:image: All the images referenced by og:image should be at least 200px in both dimensions. Please check all the images with tag og:image in the given url and ensure that it meets the recommended specification.

Object Properties

fb:app_id: 243138589129142
 og:url: <http://www.facciamoadesso.it/notizie/e-online-la-guida-per-le-imprese-colpite-dal-terremoto>
 og:type: website
 og:title: È online la guida per le imprese colpite dal terremoto
 og:image: 
 og:description: Nel documento i principali provvedimenti legati all'agibilità, all'accertamento dei danni, al riconoscimento delle spese sostenute
 og:site_name: Facciamo
 og:updated_time: 1350128219

Raw Open Graph Document Information

```

Meta Tag: <meta property="fb:app_id" content="243138589129142" />
Meta Tag: <meta property="og:url" content="http://www.facciamoadesso.it/notizie/e-online-la-guida-per-le-imprese-colpite-dal-terremoto" />
Meta Tag: <meta property="og:type" content="website" />
Meta Tag: <meta property="og:title" content="È online la guida per le imprese colpite dal terremoto " />
Meta Tag: <meta property="og:description" content="Nel documento i principali provvedimenti legati all'agibilità, all'accertamento dei danni, al riconoscimento delle spese sostenute " />
Meta Tag: <meta property="og:site_name" content="Facciamo" />
Meta Tag: <meta property="og:image" content="http://www.facciamoadesso.it/notizie/e-online-la-guida-per-le-imprese-colpite-dal-terremoto/image_mini" />
    
```

URLs

Graph API: <https://graph.facebook.com/10151150551566189>
 Scraped URL: See exactly what our scraper sees for your URL

Please, note all of those ‘og:xxx’ properties that has to be added to a standard Plone page.

There is also a warning about the image loaded in that news. It’s about the fact that the usual image_mini is 140x200 and Facebook wants to have at least one 200x200 image set on the ‘og:image’ property.

1.1.3 Understanding and configuring Facebook permissions

XXX

1.1.4 TO DO

- Change og metadatas of a ATstandard to use it in Facebook in a custom way
- Custom Archetypes

1.2 The Plone Side

1.2.1 collective.fb

This package contains `collective.opengraph`. It adds the opengraph metadata to your HTML head section.

In the future it will include also the PAS plugin to authenticate users using their Facebook's account.

Supported metadata:

- `og:site_name` name of the site
- `og:url` url of the webpage
- `og:title` title of the webpage
- `og:description` description of the webpage
- `og:image` your webpage image (either 'image' field of your context or `collective.contentleadimage` one)

from the control panel you can manage following settings: - the default `og:type` - which content types should be opengraph metadata aware

How to install it

Add `collective.fb` to your buildout's list of eggs such as:

```
[buildout]
...
eggs =
    collective.fb
...
```

and run the buildout. Start Zope, go to Site Setup -> Add-on Products in your Plone site and install the 'collective.fb' product.

How to configure it

Go to "Site Setup" -> "Add on configuration" -> "Collective Opengraph". The first two information you have to enter are:

- the **Facebook app id**
- the **Facebook app secret**

You can find it in the basic setting of your Facebook app.

Then you have to enter the **app's admins**. You can specify more than one admin as a comma separated list of names. Admins can be both a Facebook user or the id.

If you prefer to use the id, you can go to: <https://developers.facebook.com/tools/explorer>. There you can obtain the id starting from an username.

For example: <https://developers.facebook.com/tools/explorer?method=GET&path=massimo.azzolini> returns:

```
{
  "id": "1306804485",
  "name": "Massimo Azzolini",
  "first_name": "Massimo",
  "last_name": "Azzolini",
  "link": "https://www.facebook.com/massimo.azzolini",
  "username": "massimo.azzolini",
  "gender": "male",
  "locale": "en_US"
}
```

all of this to obtain an additional metadata:

```
<meta property="fb:admins" content="{YOUR_FACEBOOK_USER_ID}"/>
```

that is needed for some kind of plugins like, for example, the Facebook **Comments**'s one.

Next two fields are about the built-in Facebook object types. See <https://developers.facebook.com/docs/opengraph/objects/builtin/> for a complete list.

You have to select the **default opengraph type** (for example 'article') and you can manage the vocabulary of that drop down list in the field **Type of contents**.

At last, you have to choose to which **Content types** all the machinery applies.

As a default none of the content types is selected.

The final result is something like:

```
<meta property="og:title" content="Welcome to Plone" />
<meta property="og:url" content="http://localhost:8080/Plone/front-page" />
<meta property="og:image" content="http://localhost:8080/Plone/logo.jpg" />
<meta property="og:site_name" content="Site" />
<meta property="og:description" content="Congratulations! You have successfully installed Plone." />
<meta property="og:type" content="article" />
<meta property="fb:admins" content="1306804485" />
<meta property="fb:app_id" content="360891567328123" />
```

and it'll appear in the **HEAD** of your **HTML** page.

How to customize the behaviour of a AT

One of the goals of this package is also to allow developers extend the default metadata definition.

It's available threw **IOpengrapMetatags** adapter:

```
from collective.opengraph.interfaces IOpengraphMetatags
from collective.opengraph.viewlets import ATMetatags

class MyATMetatags(ATMetatags):

    implements(IOpengrapMetatags)

    @property
    def metatags(self):
        tags = super(MyATMetatags, self).metatags
        tags.update({'og:newtype': 'custom value'})
        return tags
```

In this way you can, for example, define all the contents as ‘webpage’ in the Site Setup configuration and add custom configuration for news as ‘articles’ and/or pages as ‘blog’.

You can also customize existing og values:

```
from collective.opengraph.interfaces IOpengraphMetatags
from collective.opengraph.viewlets import ATMetatags

class AnotherMetatags(ATMetatags):

    implements(IOpengrapMetatags)

    @property
    def title(self):
        return '%s - Lorem ipsum' % self.context.Title()
```

In this case, for example, you can manipulate the images of a page to fit the Facebook’s restriction.

how to add extra metadatas to an AT

XXX

1.2.2 Authenticate Facebook users

This package has to be refactored.

How to install it

How to configure it

OTHER

2.1 Glossary

This is a glossary for some definitions used in this documentation and still heavily under construction.

OpenGraph See:

- the Facebook documentation.
- the Open Graph protocol.