
Instagram PHP Documentation

Release 0.1.0

Marvin Osswald

Feb 12, 2018

Contents

1	Overview	3
1.1	Requirements	3
1.2	Installation	3
1.3	License	4
1.4	Contributing	4
2	Quickstart	5
2.1	Access Instagram API	5
3	Endpoints	7
3.1	Media	7
3.2	Users	10
3.3	Locations	12
3.4	Relationships	14
3.5	Tags	16
3.6	Comments & Likes	17
4	Indices and tables	19

Contents:

1.1 Requirements

1. PHP 5.6 or higher
2. Instagram App

Note: Right now one needs an Access Token from Instagram, getting it just by the SDK isn't implemented yet.

1.2 Installation

The recommended way to install Instagram PHP is with [Composer](#). Composer is a dependency management tool for PHP that allows you to declare the dependencies your project needs and installs them into your project.

```
# Install Composer
curl -sS https://getcomposer.org/installer | php
```

You can add Instagram PHP as a dependency using the composer.phar CLI:

```
php composer.phar require marvinosswald/instagram-php:~1.0
```

Alternatively, you can specify Instagram PHP as a dependency in your project's existing composer.json file:

```
{
  "require": {
    "marvinosswald/instagram-php": "~1.0"
  }
}
```

After installing, you need to require Composer's autoloader:

```
require 'vendor/autoload.php';
```

You can find out more on how to install Composer, configure autoloading, and other best-practices for defining dependencies at getcomposer.org.

1.3 License

Licensed using the [MIT license](#).

Copyright (c) 2016 Marvin Osswald <<https://github.com/marvinosswald>>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.4 Contributing

Always welcome !

This page provides a quick introduction to Instagram PHP and introductory examples. If you have not already installed, Instagram PHP, head over to the *Installation* page.

2.1 Access Instagram API

You can send requests with the SDK using a `marvinosswald\Instagram\Instagram` object.

2.1.1 Creating a Client

```
use marvinosswald\Instagram\Instagram;

$params = [
    'accessToken' => 'your-access-token',
    'clientId' => 'your-client-id',
    'clientSecret' => 'your-client-secret',
    'redirectUri' => 'your-redirect-uri'
]

$config = [
    'allow_redirects' => false
    'http_errors' => false
    ...
];

$instagram = new Instagram($params, $config);
```

The client constructor accepts two arrays :

- \$params
- \$config who can contain all GuzzleHttpClient request options see : <http://docs.guzzlephp.org/en/stable/request-options.html>

2.1.2 Access Media Endpoint

```
$media = $instagram->media (MEDIA_ID) ->get ();
```

3.1 Media

3.1.1 Constructor(\$id)

Initialize a new Media Class with an Id or empty,

```
$media = $instagram->media();  
  
// with id set:  
$media = $instagram->media(123);
```

Return

Media Returns new instance of Media Class

3.1.2 Properties

- **id** int Media_Id
- **data** StdClass **protected** as soon as the object is synced with Instagram by `get()` this property holds the response data

If loaded you can access all properties send by Instagram like:

```
$username = $media->user->username
```

3.1.3 Methods

get(\$id)

Saves the result to the public parameter `data`

Parameter

- **id** (optional) only if not already set at initialization

```
$media->get();
```

Return

`Media` Returns itself

getByShortcode(\$shortcode)

Saves the result to the public parameter `data`

Parameter

- **shortcode**

```
$media->getByShortcode($shortcode);
```

Return

`Media` Returns itself

search(\$latitude,\$longitude,\$distance)

Parameter

- **latitude**
- **longitude**
- **distance** (optional)

```
$media->search($latitude,$longitude,$distance);
```

Return

`Array` Returns array of Media Objects

comments()

```
$media->comments();
```

Return

Array Returns array of Comment Objects

comment(\$text)

Comments on the media object

- **text** Comment text

```
$media->comment($text);
```

Return

StdClass Returns Comment Object

deleteComment(\$id)

Delete comment by its comment id

- **\$id** Comment Id

```
$media->deleteComment($id);
```

likes()

Parameter

```
$media->likes();
```

Return

Array Returns array of Like Objects

like()

Likes this media Object by the current authorized user

Parameter

```
$media->like();
```

Return

StdClass Returns Like Object

unlike()

Removes like from this media Object by the current authorized user

Parameter

```
$media->unlike();
```

Return

StdClass Returns Like Object

3.2 Users

3.2.1 Constructor(\$id)

Initialize a new User Class with an Id or empty,

```
$user = $instagram->user();  
  
// with id set:  
$user = $instagram->user(123);
```

Return

User Returns new instance of User Class

3.2.2 Properties

- **id** int User_Id
- **data** StdClass **protected** as soon as the object is synced with Instagram by `get()` this property holds the response data

If loaded you can access all properties send by Instagram like:

```
$username = $user->username
```

3.2.3 Methods

get(\$id)

Properties

- **id** only if not already set at initialization

Return

User Returns itself

self()

Access authorized user

Return

User Returns itself

getMediaRecent(\$count,\$minId,\$maxId)

Requests recent media objects posted by the specified user

Parameter

- **count** (optional)
- **minId** (optional)
- **maxId** (optional)

```
$user->getMediaRecent ($count, $minId, $maxId) ;
```

Return

Array Returns array of Media Objects

selfMediaRecent(\$count,\$minId,\$maxId)

Requests recent media objects posted by the authorized user

Parameter

- **count** (optional)
- **minId** (optional)
- **maxId** (optional)

```
$user->selfMediaRecent($count,$minId,$maxId);
```

Return

Array Returns array of User Objects

selfMediaLiked(\$count,\$maxLiked)

Requests liked media objects posted by the authorized user

Parameter

- **count** (optional)
- **maxLikeId** (optional)

```
$user->getMediaLiked($count,$maxLikeId);
```

Return

Array Returns array of User Objects

search(\$query,\$count)

Properties

- **query** Search query
- **count** (optional) Limit result

Return

Array Returns array of User Objects

3.3 Locations

3.3.1 Constructor(\$id)

Initialize a new Location Class with an Id or empty,


```
$location = $instagram->location();  
  
// with id set:  
$location = $instagram->location(123);
```

Return

`Location` Returns new instance of Location Class

3.3.2 Properties

- **id** `int` `Location_Id`
- **data** `StdClass` **protected** as soon as the object is synced with Instagram by `get()` this property holds the response data

If loaded you can access all properties send by Instagram like:

```
$latitude = $location->latitude
```

3.3.3 Methods

`get($id)`

Parameter

- **id** only if not already set at initialization

Return

`Location` Returns itself

`recentMedia($minTagId,$maxTagId)`

Properties

- **minTagId** (optional)
- **maxTagId** (optional)

Return

`Array` Returns array of Media Objects

searchByCoordinates(\$lat,\$lng,\$distance)

Properties

- **latitude**
- **longitude**
- **distance** (optional) Limit radius, default is 500m

Return

Array Returns array of Location Objects

searchByFbPlaces(\$fb_places_id,\$distance)

Properties

- **fb_places_id** A Place id issued by Facebook
- **distance** (optional) Limit radius, default is 500m

Return

Array Returns array of Location Objects

3.4 Relationships

3.4.1 Constructor()

Initialize a new Relationship Class,

```
$relationship = $instagram->relationship();
```

Return

Relationship Returns new instance of Relationship Class

3.4.2 Methods

follows()

Returns User the authorized User is following

Return

Array Returns array of User Objects

followers()

Returns User who are following the authorized User

Return

Array Returns array of User Objects

followingRequests()

Returns all open following requests.

with()

Get information about a relationship to another user. Relationships are expressed using the following terms in the response: outgoing_status: Your relationship to the user. Can be 'follows', 'requested', 'none'. incoming_status: A user's relationship to you. Can be 'followed_by', 'requested_by', 'blocked_by_you', 'none'.

follow(\$target)**Parameters**

- **target** Id of user to follow

unfollow(\$target)**Parameters**

- **target** Id of user not to follow anymore

approve(\$target)**Parameters**

- **target** Id of user to approve request

ignore(\$target)**Parameters**

- **target** Id of user to ignore request

3.5 Tags

3.5.1 Constructor(\$tagname)

3.5.2 Properties

- **id** `int` Tag name
- **data** `stdClass` **protected** as soon as the object is synced with Instagram by `get()` this property holds the response data

If loaded you can access all properties send by Instagram like:

```
$media_count = $tag->media_count
```

Initialize a new Tag Class with an Tagname or empty,

```
$tag = $instagram->tag();  
  
// with id set:  
$tag = $instagram->tag(123);
```

Return

Tag Returns new instance of Tag Class

3.5.3 Methods

get(\$id)

Parameter

- **id** only if not already set at initialization

Return

Tag Returns itself

recentMedia(\$count,\$minTagId,\$maxTagId)

Parameter

- **count** (optional)
- **minTagId** (optional)
- **maxTagId** (optional)

Return

Array Returns array of Media objects

search(\$query)

Parameter

- **query** Search query

Return

Array of Tag Objects

3.6 Comments & Likes

Both of these Endpoints are embedded into the Media Class

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`