

# API Reference

# Instagram

API Version: 1.0.0

Description of Instagram RESTful API.

Current limitations:

- Instagram service does not support [cross origin headers]([https://developer.mozilla.org/en-US/docs/Web/HTTP/Access\\_control\\_CORS](https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS))
- for security reasons, therefore it is not possible to use Swagger UI and make API calls directly from browser.
- Modification API requests (POST, DELETE) require additional security [scopes](<https://instagram.com/developer/authorization/>)
- that are available for Apps [created on or after Nov 17, 2015](<http://instagram.com/developer/review/>) and
- started in [Sandbox Mode](<http://instagram.com/developer/sandbox/>).
- Consider the [Instagram limitations](<https://instagram.com/developer/limits/>) for API calls that depends on App Mode.

Warning: For Apps [created on or after Nov 17, 2015](<http://instagram.com/developer/changelog/>) API responses containing media objects no longer return the `data` field in `comments` and `likes` nodes.

Last update: 2015-11-28

## CONTACT

**NAME:** Instagram

**URL:** <https://instagram.com/developer/support>

**Terms of service:** <https://instagram.com/about/legal/terms/api/>

# INDEX

<b>1. COMMENTS</b>	<b>4</b>
1.1 GET /media/{media_id}/comments	4
1.2 POST /media/{media_id}/comments	4
1.3 DELETE /media/{media_id}/comments/{comment_id}	5
<b>2. GEOGRAPHIES</b>	<b>6</b>
2.1 GET /geographies/{geo_id}/media/recent	6
<b>3. LIKES</b>	<b>9</b>
3.1 GET /media/{media_id}/likes	9
3.2 POST /media/{media_id}/likes	9
3.3 DELETE /media/{media_id}/likes	10
<b>4. LOCATIONS</b>	<b>11</b>
4.1 GET /locations/search	11
4.2 GET /locations/{location_id}	11
4.3 GET /locations/{location_id}/media/recent	12
<b>5. MEDIA</b>	<b>15</b>
5.1 GET /media/popular	15
5.2 GET /media/search	17
5.3 GET /media/shortcode/{shortcode}	19
5.4 GET /media/{media_id}	21
<b>6. RELATIONSHIPS</b>	<b>25</b>
6.1 GET /users/self/requested-by	25
6.2 GET /users/{user_id}/followed-by	25
6.3 GET /users/{user_id}/follows	26
6.4 GET /users/{user_id}/relationship	26
6.5 POST /users/{user_id}/relationship	27
<b>7. TAGS</b>	<b>28</b>
7.1 GET /tags/search	28
7.2 GET /tags/{tag_name}	28
7.3 GET /tags/{tag_name}/media/recent	29
<b>8. USERS</b>	<b>32</b>
8.1 GET /users/search	32
8.2 GET /users/self/feed	32
8.3 GET /users/self/media/liked	35
8.4 GET /users/{user_id}	37
8.5 GET /users/{user_id}/media/recent	38

# Security and Authentication

## SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
api_key	apiKey	
instagram_auth	oauth2	

# API

## 1. COMMENTS

Manage comments

### 1.1 GET /media/{media\_id}/comments

Get a list of recent comments on a media object.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

#### RESPONSE

STATUS CODE - 200: List of comments of the media resource.

##### RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    created_time string Comment creation UNIX timestamp
    from {
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }
    id string ID of this comment
    text string Text of the comment
  }]
  meta {
    code integer HTTP result code
  }
}
```

### 1.2 POST /media/{media\_id}/comments

Create a comment on a media object.

Create a comment on a media object with the following rules:

- The total length of the comment cannot exceed 300 characters.
- The comment cannot contain more than 4 hashtags.
- The comment cannot contain more than 1 URL.
- The comment cannot consist of all capital letters.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
------	------	---------	-------------

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

## QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*text	string		Text to post as a comment on the media object as specified in `media-id`.

## RESPONSE

STATUS CODE - 200: Result of posting a comment.

RESPONSE MODEL - application/json

```
{
  data string No data - 'null'
  meta {
    code integer HTTP result code
  }
}
```

### 1.3 DELETE /media/{media\_id}/comments/{comment\_id}

**Remove a comment.**

Remove a comment either on the authenticated user's media object or authored by the authenticated user.

## REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.
*comment_id	string		The ID of the comment entry.

## RESPONSE

STATUS CODE - 200: Result of deleting a comment.

RESPONSE MODEL - application/json

```
{
  data string No data - 'null'
  meta {
    code integer HTTP result code
  }
}
```

## 2. GEOGRAPHIES

Get geographies

### 2.1 GET /geographies/{geo\_id}/media/recent

Get recent media from a custom geo-id.

Get recent media from a geography subscription that you created.

Note: You can only access Geographies that were explicitly created by your OAuth client. Check the Geography Subscriptions section of the [real-time updates page](https://instagram.com/developer/realtime/). When you create a subscription to some geography that you define, you will be returned a unique `geo-id` that can be used in this query. To backfill photos from the location covered by this geography, use the [media search endpoint](https://instagram.com/developer/endpoints/media/).

Warning: [Deprecated](http://instagram.com/developer/changelog/) for Apps created on or after Nov 17, 2015

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*geo_id	string		The geography ID.

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
count	int32		Max number of media to return.
min_id	string		Return media before this `min_id`.

### RESPONSE

STATUS CODE - 200: List of recent media entries from a geography subscription.

RESPONSE MODEL - application/json

```
{
  data [{
    attribution      string      ??? Unknown ???
    caption {
      created_time  string      Caption creation UNIX timestamp
      from {
        full_name    string      User full name
        id            string      User ID
        profile_picture string    URL to user profile picture
        username      string      User name, nickname
      }
      id            string      ID of this caption
      text          string      Caption text
    }
    comments {
      count integer    Number of comments available, data does not necessary contain all comments
      data [{
        Array of object:
      }
    ]
  }
}
```

```

    created_time string Comment creation UNIX timestamp
    from {
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }
    id string ID of this comment
    text string Text of the comment
  ]]
}
created_time string Media creation UNIX timestamp
filter string Filter of this media entry
id string ID of a media entry
images {
  low_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  standard_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  thumbnail {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
}
likes {
  count integer Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }]
}
link string Fixed URL of this media entry
location {
  id string ID of this location (in some responses it has a type of 'integer')
  latitude number Location latitude
  longitude number Location longitude
  name string Location name
}
tags [string]
type enum ALLOWED:image, video
Type of this media entry
user {
  full_name string User full name
  id string User ID
  profile_picture string URL to user profile picture
  username string User name, nickname
}

```

**user\_has\_liked** **boolean** Indicates whether authenticated user has liked this media or not

**users\_in\_photo** [{

Array of object:

**position** {

**x** **number** X position (horizontal)

**y** **number** Y position (vertical)

}

**user** {

**full\_name** **string** User full name

**id** **string** User ID

**profile\_picture** **string** URL to user profile picture

**username** **string** User name, nickname

}

}]

**videos** {

**low\_resolution** {

**height** **integer** Image/video height in pixels

**url** **string** URL of the image/video resource

**width** **integer** Image/video width in pixels

}

**standard\_resolution** {

**height** **integer** Image/video height in pixels

**url** **string** URL of the image/video resource

**width** **integer** Image/video width in pixels

}

}

}]

**meta** {

**code** **integer** HTTP result code

}

**pagination** {

**next\_max\_id** **string** The max ID of the next page

**next\_url** **string** URL to retrieve next page of entries

}

}



## 3. LIKES

Manage likes

### 3.1 GET /media/{media\_id}/likes

Get a list of users who have liked this media.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

#### RESPONSE

STATUS CODE - 200: List of users who liked the media resource.

##### RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    full_name      string  User full name
    id             string  User ID
    profile_picture string  URL to user profile picture
    username       string  User name, nickname
  }]
  meta {
    code integer HTTP result code
  }
}
```

### 3.2 POST /media/{media\_id}/likes

Set a like on this media by the current user.

Set a like on this media by the currently authenticated user.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

#### RESPONSE

STATUS CODE - 200: Result of setting a like.

##### RESPONSE MODEL - application/json

```
{
  data string  No data - 'null'
  meta {
```

```
    code integer HTTP result code
  }
}
```

---

### 3.3 DELETE /media/{media\_id}/likes

**Remove a like on this media by the current user.**

Remove a like on this media by the currently authenticated user.

#### REQUEST

##### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

---

#### RESPONSE

**STATUS CODE - 200:** Result of removing a like.

**RESPONSE MODEL - application/json**

```
{
  data string No data - 'null'
  meta {
    code integer HTTP result code
  }
}
```

---

# 4. LOCATIONS

Get information about locations

## 4.1 GET /locations/search

Search for a location by geographic coordinate.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
distance	int32		Default is 1000m (distance=1000), max distance is 5000.
facebook_places_id	string		Returns a location mapped off of a Facebook places id. If used, a Foursquare id and `lat`, `lng` are not required.
foursquare_id	string		Returns a location mapped off of a foursquare v1 api location id. If used, you are not required to use `lat` and `lng`. Note that this method is deprecated; you should use the new foursquare IDs with V2 of their API.
lat	double		Latitude of the center search coordinate. If used, `lng` is required.
lng	double		Longitude of the center search coordinate. If used, `lat` is required.
foursquare_v2_id	string		Returns a location mapped off of a foursquare v2 api location id. If used, you are not required to use `lat` and `lng`.

### RESPONSE

STATUS CODE - 200: List of found locations.

#### RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    id          string  ID of this location (in some responses it has a type of 'integer')
    latitude    number  Location latitude
    longitude   number  Location longitude
    name        string  Location name
  }]
  meta {
    code integer HTTP result code
  }
}
```

## 4.2 GET /locations/{location\_id}

Get information about a location.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
------	------	---------	-------------

NAME	TYPE	EXAMPLE	DESCRIPTION
*location_id	string		The location ID.

## RESPONSE

STATUS CODE - 200: Location information response.

RESPONSE MODEL - application/json

```
{
  data {
    id          string  ID of this location (in some responses it has a type of 'integer')
    latitude    number  Location latitude
    longitude   number  Location longitude
    name        string  Location name
  }
  meta {
    code integer HTTP result code
  }
}
```

### 4.3 GET /locations/{location\_id}/media/recent

Get a list of recent media objects from a given location.

## REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*location_id	string		The location ID.

QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
min_timestamp	int64		Return media after this UNIX timestamp.
max_timestamp	int64		Return media before this UNIX timestamp.
min_id	string		Return media before this `min_id`.
max_id	string		Return media after this `max_id`.

## RESPONSE

STATUS CODE - 200: List of media entries from this location.

RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    attribution    string    ??? Unknown ???
    caption {
      created_time string    Caption creation UNIX timestamp
      from {
        full_name   string    User full name
      }
    }
  }
}
```

```

    id            string    User ID
    profile_picture string    URL to user profile picture
    username      string    User name, nickname
  }
  id            string    ID of this caption
  text         string    Caption text
}
comments {
  count integer    Number of comments available, data does not necessary contain all comments
  data [{
    Array of object:
      created_time string    Comment creation UNIX timestamp
      from {
        full_name      string    User full name
        id             string    User ID
        profile_picture string    URL to user profile picture
        username       string    User name, nickname
      }
      id            string    ID of this comment
      text         string    Text of the comment
    }]
}
created_time    string    Media creation UNIX timestamp
filter         string    Filter of this media entry
id            string    ID of a media entry
images {
  low_resolution {
    height integer    Image/video height in pixels
    url      string    URL of the image/video resource
    width   integer    Image/video width in pixels
  }
  standard_resolution {
    height integer    Image/video height in pixels
    url      string    URL of the image/video resource
    width   integer    Image/video width in pixels
  }
  thumbnail {
    height integer    Image/video height in pixels
    url      string    URL of the image/video resource
    width   integer    Image/video width in pixels
  }
}
likes {
  count integer    Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
      full_name      string    User full name
      id             string    User ID
      profile_picture string    URL to user profile picture
      username       string    User name, nickname
    }]
}
link           string    Fixed URL of this media entry
location {
  id           string    ID of this location (in some responses it has a type of 'integer')
  latitude     number    Location latitude
  longitude    number    Location longitude
}

```



# 5. MEDIA

Get information about media

## 5.1 GET /media/popular

Get a list of currently popular media.

Get a list of what media is most popular at the moment. Can return mix of `image` and `video` types.

Warning: [Deprecated](<http://instagram.com/developer/changelog/>) for Apps created on or after Nov 17, 2015

### REQUEST

No request parameters

### RESPONSE

STATUS CODE - 200: Found media resources (without likes information).

RESPONSE MODEL - application/json

```
{
  data [{
    attribution      string      ??? Unknown ???
    caption {
      created_time  string      Caption creation UNIX timestamp
      from {
        full_name    string      User full name
        id            string      User ID
        profile_picture string    URL to user profile picture
        username      string      User name, nickname
      }
      id            string      ID of this caption
      text          string      Caption text
    }
    comments {
      count integer    Number of comments available, data does not necessary contain all comments
      data [{
        created_time string    Comment creation UNIX timestamp
        from {
          full_name    string    User full name
          id            string    User ID
          profile_picture string  URL to user profile picture
          username      string    User name, nickname
        }
        id            string    ID of this comment
        text          string    Text of the comment
      }]
    }
  }
  created_time      string      Media creation UNIX timestamp
  filter            string      Filter of this media entry
  id                string      ID of a media entry
  images {
    low_resolution {
      height integer    Image/video height in pixels
```

```

    url    string    URL of the image/video resource
    width  integer    Image/video width in pixels
}
standard_resolution {
    height integer    Image/video height in pixels
    url    string    URL of the image/video resource
    width  integer    Image/video width in pixels
}
thumbnail {
    height integer    Image/video height in pixels
    url    string    URL of the image/video resource
    width  integer    Image/video width in pixels
}
}
likes {
    count integer    Number of likes available, data does not necessary contain all comments
    data [{
    Array of object:
        full_name    string    User full name
        id            string    User ID
        profile_picture string    URL to user profile picture
        username      string    User name, nickname
    }]
}
link            string    Fixed URL of this media entry
location {
    id            string    ID of this location (in some responses it has a type of 'integer')
    latitude      number    Location latitude
    longitude     number    Location longitude
    name          string    Location name
}
tags            [string]
type            enum      ALLOWED:image, video
                    Type of this media entry
user {
    full_name    string    User full name
    id            string    User ID
    profile_picture string    URL to user profile picture
    username      string    User name, nickname
}
user_has_liked boolean    Indicates whether authenticated user has liked this media or not
users_in_photo [{
    Array of object:
        position {
            x number    X position (horizontal)
            y number    Y position (vertical)
        }
        user {
            full_name    string    User full name
            id            string    User ID
            profile_picture string    URL to user profile picture
            username      string    User name, nickname
        }
    }]
}
videos {
    low_resolution {
        height integer    Image/video height in pixels

```



```

        url    string  URL of the image/video resource
        width  integer  Image/video width in pixels
    }
    standard_resolution {
        height integer  Image/video height in pixels
        url    string  URL of the image/video resource
        width  integer  Image/video width in pixels
    }
}
}]
meta {
    code integer  HTTP result code
}
}

```

## 5.2 GET /media/search

### Search for media in a given area.

Search for media in a given area. The default time span is set to 5 days. The time span must not exceed 7 days. Defaults time stamps cover the last 5 days. Can return mix of `image` and `video` types.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*lat	double		Latitude of the center search coordinate. If used, `lng` is required.
*lng	double		Longitude of the center search coordinate. If used, `lat` is required.
min_timestamp	int64		A unix timestamp. All media returned will be taken later than this timestamp.
max_timestamp	int64		A unix timestamp. All media returned will be taken earlier than this timestamp.
distance	int32		Default is 1km (distance=1000), max distance is 5km.

### RESPONSE

STATUS CODE - 200: Found media resources (without likes information) in a given area.

#### RESPONSE MODEL - application/json

```

{
  data [{
    attribution    string    ??? Unknown ???
    caption {
      created_time string    Caption creation UNIX timestamp
      from {
        full_name   string    User full name
        id          string    User ID
        profile_picture string  URL to user profile picture
        username    string    User name, nickname
      }
      id          string    ID of this caption
      text        string    Caption text
    }
    comments {
      count integer  Number of comments available, data does not necessary contain all comments
    }
  }
}

```

```

data [{
  Array of object:
    created_time string Comment creation UNIX timestamp
    from {
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }
    id string ID of this comment
    text string Text of the comment
  }]
}
created_time string Media creation UNIX timestamp
filter string Filter of this media entry
id string ID of a media entry
images {
  low_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  standard_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  thumbnail {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
}
likes {
  count integer Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }]
}
link string Fixed URL of this media entry
location {
  id string ID of this location (in some responses it has a type of 'integer')
  latitude number Location latitude
  longitude number Location longitude
  name string Location name
}
tags [string]
type enum ALLOWED:image, video
Type of this media entry
user {
  full_name string User full name
  id string User ID
  profile_picture string URL to user profile picture
}

```



```

data {
  attribution      string      ??? Unknown ???
  caption {
    created_time  string      Caption creation UNIX timestamp
    from {
      full_name    string      User full name
      id            string      User ID
      profile_picture string    URL to user profile picture
      username      string      User name, nickname
    }
    id              string      ID of this caption
    text            string      Caption text
  }
  comments {
    count integer    Number of comments available, data does not necessary contain all comments
    data [{
      Array of object:
        created_time string    Comment creation UNIX timestamp
        from {
          full_name    string    User full name
          id            string    User ID
          profile_picture string  URL to user profile picture
          username      string    User name, nickname
        }
        id              string    ID of this comment
        text            string    Text of the comment
      }]
  }
  created_time    string      Media creation UNIX timestamp
  filter          string      Filter of this media entry
  id              string      ID of a media entry
  images {
    low_resolution {
      height integer    Image/video height in pixels
      url      string    URL of the image/video resource
      width   integer    Image/video width in pixels
    }
    standard_resolution {
      height integer    Image/video height in pixels
      url      string    URL of the image/video resource
      width   integer    Image/video width in pixels
    }
    thumbnail {
      height integer    Image/video height in pixels
      url      string    URL of the image/video resource
      width   integer    Image/video width in pixels
    }
  }
}
likes {
  count integer    Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
      full_name    string    User full name
      id            string    User ID
      profile_picture string  URL to user profile picture
      username      string    User name, nickname
    }]
}

```

```

}
link          string      Fixed URL of this media entry
location {
  id          string      ID of this location (in some responses it has a type of 'integer')
  latitude    number      Location latitude
  longitude    number      Location longitude
  name        string      Location name
}
tags          [string]
type          enum        ALLOWED:image, video
                                Type of this media entry
user {
  full_name   string      User full name
  id          string      User ID
  profile_picture string  URL to user profile picture
  username    string      User name, nickname
}
user_has_liked boolean    Indicates whether authenticated user has liked this media or not
users_in_photo [{
Array of object:
  position {
    x number  X position (horizontal)
    y number  Y position (vertical)
  }
  user {
    full_name   string  User full name
    id          string  User ID
    profile_picture string  URL to user profile picture
    username    string  User name, nickname
  }
}]
videos {
  low_resolution {
    height integer  Image/video height in pixels
    url    string  URL of the image/video resource
    width  integer  Image/video width in pixels
  }
  standard_resolution {
    height integer  Image/video height in pixels
    url    string  URL of the image/video resource
    width  integer  Image/video width in pixels
  }
}
}
meta {
  code integer  HTTP result code
}
}

```

## 5.4 GET /media/{media\_id}

### Get information about a media object.

Get information about a media object. The returned type key will allow you to differentiate between image and video media.

Note: if you authenticate with an OAuth Token, you will receive the user\_has\_liked key which quickly tells

you whether the current user has liked this media item.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*media_id	string		The ID of the media resource.

## RESPONSE

STATUS CODE - 200: Media resource information.

### RESPONSE MODEL - application/json

```
{
  data {
    attribution      string      ??? Unknown ???
    caption {
      created_time  string      Caption creation UNIX timestamp
      from {
        full_name    string      User full name
        id            string      User ID
        profile_picture string    URL to user profile picture
        username      string      User name, nickname
      }
      id            string      ID of this caption
      text          string      Caption text
    }
    comments {
      count integer    Number of comments available, data does not necessary contain all comments
      data [{
        Array of object:
          created_time string    Comment creation UNIX timestamp
          from {
            full_name    string    User full name
            id            string    User ID
            profile_picture string  URL to user profile picture
            username      string    User name, nickname
          }
          id            string    ID of this comment
          text          string    Text of the comment
        }]
    }
  }
  created_time      string      Media creation UNIX timestamp
  filter            string      Filter of this media entry
  id                string      ID of a media entry
  images {
    low_resolution {
      height integer    Image/video height in pixels
      url      string    URL of the image/video resource
      width   integer    Image/video width in pixels
    }
    standard_resolution {
      height integer    Image/video height in pixels
      url      string    URL of the image/video resource
      width   integer    Image/video width in pixels
    }
  }
}
```

```

    thumbnail {
      height integer Image/video height in pixels
      url string URL of the image/video resource
      width integer Image/video width in pixels
    }
  }
  likes {
    count integer Number of likes available, data does not necessary contain all comments
    data [{
      Array of object:
        full_name string User full name
        id string User ID
        profile_picture string URL to user profile picture
        username string User name, nickname
      }]
  }
  link string Fixed URL of this media entry
  location {
    id string ID of this location (in some responses it has a type of 'integer')
    latitude number Location latitude
    longitude number Location longitude
    name string Location name
  }
  tags [string]
  type enum ALLOWED:image, video
      Type of this media entry
  user {
    full_name string User full name
    id string User ID
    profile_picture string URL to user profile picture
    username string User name, nickname
  }
  user_has_liked boolean Indicates whether authenticated user has liked this media or not
  users_in_photo [{
    Array of object:
      position {
        x number X position (horizontal)
        y number Y position (vertical)
      }
      user {
        full_name string User full name
        id string User ID
        profile_picture string URL to user profile picture
        username string User name, nickname
      }
    }]
  videos {
    low_resolution {
      height integer Image/video height in pixels
      url string URL of the image/video resource
      width integer Image/video width in pixels
    }
    standard_resolution {
      height integer Image/video height in pixels
      url string URL of the image/video resource
      width integer Image/video width in pixels
    }
  }

```

```
    }  
  }  
  meta {  
    code integer HTTP result code  
  }  
}
```

---



# 6. RELATIONSHIPS

Relationships to other users outgoing and incoming

## 6.1 GET /users/self/requested-by

List the users who have requested this user's permission to follow.

### REQUEST

No request parameters

### RESPONSE

**STATUS CODE - 200:** List of users who have requested this user's permission to follow.

**RESPONSE MODEL - application/json**

```
{
  data [{
    Array of object:
    full_name      string  User full name
    id             string  User ID
    profile_picture string  URL to user profile picture
    username       string  User name, nickname
  }]
  meta {
    code integer HTTP result code
  }
}
```

---

## 6.2 GET /users/{user\_id}/followed-by

Get the list of users this user is followed by.

Get the list of users this user is followed by. To get users followed by the owner of the access token, you can use self instead of the user\_id.

### REQUEST

**PATH PARAMETERS**

NAME	TYPE	EXAMPLE	DESCRIPTION
*user_id	string		The ID of a user, or <b>**self**</b> to retrieve information about authenticated user.

### RESPONSE

**STATUS CODE - 200:** List of users this user is followed by.

**RESPONSE MODEL - application/json**

```
{
  data [{
    Array of object:
    full_name      string  User full name
    id             string  User ID
    profile_picture string  URL to user profile picture
    username       string  User name, nickname
  }]
}
```

```

    ]]
    meta {
      code integer HTTP result code
    }
    pagination {
      next_cursor string The cursor ID of the next page
      next_url string URL to retrieve next page of entries
    }
  }
}

```

## 6.3 GET /users/{user\_id}/follows

Get the list of users this user follows.

Get the list of users this user follows. To get follows of the owner of the access token, you can use self instead of the user\_id.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*user_id	string		The ID of a user, or <b>**self**</b> to retrieve information about authenticated user.

### RESPONSE

STATUS CODE - 200: List of users this user follows.

RESPONSE MODEL - application/json

```

{
  data [{
    Array of object:
    full_name string User full name
    id string User ID
    profile_picture string URL to user profile picture
    username string User name, nickname
  }]
  meta {
    code integer HTTP result code
  }
  pagination {
    next_cursor string The cursor ID of the next page
    next_url string URL to retrieve next page of entries
  }
}

```

## 6.4 GET /users/{user\_id}/relationship

Get information about a relationship to another user.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*user_id	string		The ID of a user to get information about.

## RESPONSE

STATUS CODE - 200: Relationship information.

RESPONSE MODEL - application/json

```
{
  data {
    incoming_status      enum      ALLOWED:none, followed_by, requested_by
                               Status of incoming relationship
    outgoing_status      enum      ALLOWED:none, follows, requested
                               Status of outgoing relationship
    target_user_is_private boolean Indicates whether target user is private or not
  }
  meta {
    code integer HTTP result code
  }
}
```

---

## 6.5 POST /users/{user\_id}/relationship

Modify the relationship between the current user and the target user.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*user_id	string		The ID of the target user.

---

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*action	enum ALLOWED: follow, unfollow, block, unblock, approve, ignore		Type of action to apply for relationship with the user.

---

## RESPONSE

STATUS CODE - 200: Relationship information.

RESPONSE MODEL - application/json

```
{
  data {
    outgoing_status      enum      ALLOWED:none, follows, requested
                               Status of outgoing relationship
  }
  meta {
    code integer HTTP result code
  }
}
```

---

# 7. TAGS

Get information about tags

## 7.1 GET /tags/search

Search for tags by name.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*q	string		A valid tag name without a leading \#. (eg. snowy, nofilter)

### RESPONSE

STATUS CODE - 200: List of found tags and their statistics.

#### RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    media_count integer Overall number of media entries tagged with this name
    name string Tag name
  }]
  meta {
    code integer HTTP result code
  }
}
```

## 7.2 GET /tags/{tag\_name}

Get information about a tag object.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*tag_name	string		The tag name.

### RESPONSE

STATUS CODE - 200: Tag information response.

#### RESPONSE MODEL - application/json

```
{
  data {
    media_count integer Overall number of media entries tagged with this name
    name string Tag name
  }
  meta {
    HTTP result code
  }
}
```

```

    code integer
  }
}

```

## 7.3 GET /tags/{tag\_name}/media/recent

### Get a list of recently tagged media.

Get a list of recently tagged media. Use the `max_tag_id` and `min_tag_id` parameters in the pagination response to paginate through these objects.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*tag_name	string		The tag name.

### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
count	integer		Count of tagged media to return.
min_tag_id	string		Return media before this `min_tag_id`.
max_tag_id	string		Return media after this `max_tag_id`.

## RESPONSE

STATUS CODE - 200: List of media entries with this tag.

### RESPONSE MODEL - application/json

```

{
  data [{
    attribution string ??? Unknown ???
    caption {
      created_time string Caption creation UNIX timestamp
      from {
        full_name string User full name
        id string User ID
        profile_picture string URL to user profile picture
        username string User name, nickname
      }
      id string ID of this caption
      text string Caption text
    }
    comments {
      count integer Number of comments available, data does not necessary contain all comments
      data [{
        created_time string Comment creation UNIX timestamp
        from {
          full_name string User full name
          id string User ID
          profile_picture string URL to user profile picture
          username string User name, nickname
        }
      }
    }
  }
}

```

```

    }
    id          string  ID of this comment
    text        string  Text of the comment
  }]
}
created_time   string  Media creation UNIX timestamp
filter         string  Filter of this media entry
id             string  ID of a media entry
images {
  low_resolution {
    height integer Image/video height in pixels
    url    string  URL of the image/video resource
    width  integer Image/video width in pixels
  }
  standard_resolution {
    height integer Image/video height in pixels
    url    string  URL of the image/video resource
    width  integer Image/video width in pixels
  }
  thumbnail {
    height integer Image/video height in pixels
    url    string  URL of the image/video resource
    width  integer Image/video width in pixels
  }
}
likes {
  count integer  Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
      full_name   string  User full name
      id          string  User ID
      profile_picture string URL to user profile picture
      username    string  User name, nickname
    }]
}
link          string  Fixed URL of this media entry
location {
  id          string  ID of this location (in some responses it has a type of 'integer')
  latitude    number  Location latitude
  longitude   number  Location longitude
  name        string  Location name
}
tags          [string]
type          enum    ALLOWED:image, video
                Type of this media entry
user {
  full_name   string  User full name
  id          string  User ID
  profile_picture string URL to user profile picture
  username    string  User name, nickname
}
user_has_liked boolean Indicates whether authenticated user has liked this media or not
users_in_photo [{
  Array of object:
    position {
      x number X position (horizontal)
      y number Y position (vertical)
    }
  }
}

```

```

    }
    user {
        full_name      string  User full name
        id              string  User ID
        profile_picture string  URL to user profile picture
        username        string  User name, nickname
    }
}
videos {
    low_resolution {
        height integer Image/video height in pixels
        url     string  URL of the image/video resource
        width  integer Image/video width in pixels
    }
    standard_resolution {
        height integer Image/video height in pixels
        url     string  URL of the image/video resource
        width  integer Image/video width in pixels
    }
}
}
}
meta {
    code integer HTTP result code
}
pagination {
    deprecation_warning string The deprecation warning, if information is available
    min_tag_id          string The min ID of a tag for the next page
    next_max_id         string Deprecated. Use min_tag_id instead
    next_max_tag_id     string The max ID of a tag for the next page
    next_min_id         string Deprecated. Use max_tag_id instead
    next_url            string URL to retrieve next page of entries
}
}

```

---

## 8. USERS

Get information about users

### 8.1 GET /users/search

Search for a user by name.

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*q	string		A query string.
count	integer		Number of users to return.

#### RESPONSE

STATUS CODE - 200: List of found users.

RESPONSE MODEL - application/json

```
{
  data [{
    full_name      string  User full name
    id             string  User ID
    profile_picture string  URL to user profile picture
    username       string  User name, nickname
  }]
  meta {
    code integer HTTP result code
  }
}
```

### 8.2 GET /users/self/feed

See the authenticated user's feed.

See the authenticated user's feed.

Warning: [Deprecated](<http://instagram.com/developer/changelog/>) for Apps created on or after Nov 17, 2015

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
count	integer		Count of media to return.
min_id	string		Return media later than this `min_id`.
max_id	string		Return media earlier than this `max_id`.

#### RESPONSE



## STATUS CODE - 200: Users feed entries.

### RESPONSE MODEL - application/json

```
{
  data [{
    attribution      string      ??? Unknown ???
    caption {
      created_time  string      Caption creation UNIX timestamp
      from {
        full_name    string      User full name
        id            string      User ID
        profile_picture string    URL to user profile picture
        username      string      User name, nickname
      }
      id            string      ID of this caption
      text          string      Caption text
    }
    comments {
      count integer    Number of comments available, data does not necessary contain all comments
      data [{
        Array of object:
          created_time string    Comment creation UNIX timestamp
          from {
            full_name    string    User full name
            id            string    User ID
            profile_picture string  URL to user profile picture
            username      string    User name, nickname
          }
          id            string    ID of this comment
          text          string    Text of the comment
        }]
      }
    created_time    string      Media creation UNIX timestamp
    filter          string      Filter of this media entry
    id              string      ID of a media entry
    images {
      low_resolution {
        height integer    Image/video height in pixels
        url    string     URL of the image/video resource
        width  integer    Image/video width in pixels
      }
      standard_resolution {
        height integer    Image/video height in pixels
        url    string     URL of the image/video resource
        width  integer    Image/video width in pixels
      }
      thumbnail {
        height integer    Image/video height in pixels
        url    string     URL of the image/video resource
        width  integer    Image/video width in pixels
      }
    }
  }
  likes {
    count integer    Number of likes available, data does not necessary contain all comments
    data [{
      Array of object:
```

```

        full_name      string  User full name
        id             string  User ID
        profile_picture string  URL to user profile picture
        username       string  User name, nickname
    }}
}
link          string  Fixed URL of this media entry
location {
    id         string  ID of this location (in some responses it has a type of 'integer')
    latitude   number  Location latitude
    longitude  number  Location longitude
    name       string  Location name
}
tags          [string]
type          enum    ALLOWED:image, video
                Type of this media entry
user {
    full_name      string  User full name
    id             string  User ID
    profile_picture string  URL to user profile picture
    username       string  User name, nickname
}
user_has_liked boolean  Indicates whether authenticated user has liked this media or not
users_in_photo [{
    Array of object:
        position {
            x number  X position (horizontal)
            y number  Y position (vertical)
        }
        user {
            full_name      string  User full name
            id             string  User ID
            profile_picture string  URL to user profile picture
            username       string  User name, nickname
        }
    }]
}
videos {
    low_resolution {
        height integer  Image/video height in pixels
        url     string  URL of the image/video resource
        width  integer  Image/video width in pixels
    }
    standard_resolution {
        height integer  Image/video height in pixels
        url     string  URL of the image/video resource
        width  integer  Image/video width in pixels
    }
}
}}
meta {
    code integer  HTTP result code
}
pagination {
    next_max_id string  The max ID of the next page
    next_url    string  URL to retrieve next page of entries
}
}

```

## 8.3 GET /users/self/media/liked

See the list of media liked by the authenticated user.

See the list of media liked by the authenticated user. Private media is returned as long as the authenticated user has permission to view that media. Liked media lists are only available for the currently authenticated user.

### REQUEST

#### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
count	integer		Count of media to return.
max_like_id	string		Return media liked before this id.

### RESPONSE

STATUS CODE - 200: Users media entries.

#### RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    attribution      string      ??? Unknown ???
    caption {
      created_time  string      Caption creation UNIX timestamp
      from {
        full_name   string      User full name
        id           string      User ID
        profile_picture string  URL to user profile picture
        username    string      User name, nickname
      }
      id            string      ID of this caption
      text          string      Caption text
    }
    comments {
      count integer  Number of comments available, data does not necessary contain all comments
      data [{
        Array of object:
        created_time string  Comment creation UNIX timestamp
        from {
          full_name   string  User full name
          id           string  User ID
          profile_picture string URL to user profile picture
          username    string  User name, nickname
        }
        id            string  ID of this comment
        text          string  Text of the comment
      }]
    }
  }
  created_time  string      Media creation UNIX timestamp
  filter        string      Filter of this media entry
  id            string      ID of a media entry
  images {
    low_resolution {
      height integer  Image/video height in pixels
      url      string  URL of the image/video resource
    }
  }
}
```

```

    width integer Image/video width in pixels
}
standard_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
}
thumbnail {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
}
}
likes {
    count integer Number of likes available, data does not necessary contain all comments
    data [{
    Array of object:
        full_name string User full name
        id string User ID
        profile_picture string URL to user profile picture
        username string User name, nickname
    }]
}
link string Fixed URL of this media entry
location {
    id string ID of this location (in some responses it has a type of 'integer')
    latitude number Location latitude
    longitude number Location longitude
    name string Location name
}
tags [string]
type enum ALLOWED:image, video
Type of this media entry
user {
    full_name string User full name
    id string User ID
    profile_picture string URL to user profile picture
    username string User name, nickname
}
user_has_liked boolean Indicates whether authenticated user has liked this media or not
users_in_photo [{
Array of object:
    position {
        x number X position (horizontal)
        y number Y position (vertical)
    }
    user {
        full_name string User full name
        id string User ID
        profile_picture string URL to user profile picture
        username string User name, nickname
    }
}]
videos {
    low_resolution {
        height integer Image/video height in pixels
        url string URL of the image/video resource

```

```

        width integer Image/video width in pixels
    }
    standard_resolution {
        height integer Image/video height in pixels
        url string URL of the image/video resource
        width integer Image/video width in pixels
    }
}
}]
meta {
    code integer HTTP result code
}
pagination {
    next_max_id string The max ID of the next page
    next_url string URL to retrieve next page of entries
}
}

```

## 8.4 GET /users/{user\_id}

### Get basic information about a user.

Get basic information about a user. To get information about the owner of the access token, you can use `self` instead of the `user_id`.

Security scope `public_content` is required to read information about other users.

### REQUEST

#### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*user_id	string		The ID of a user to get information about, or <code>**self**</code> to retrieve information about authenticated user.

### RESPONSE

STATUS CODE - 200: User basic information.

#### RESPONSE MODEL - application/json

```

{
  data {
    bio string User biography
    counts {
      followed_by integer Number of followers of this user
      follows integer Number of users followed by this user
      media integer Number of user media
    }
    full_name string User full name
    id string User ID
    profile_picture string URL to user profile picture
    username string User name, nickname
    website string URL to user web-site
  }
  meta {
    code integer HTTP result code
  }
}

```

```
}  
}
```

STATUS CODE - 404: Not Found, user with such ID does not exist.

## 8.5 GET /users/{user\_id}/media/recent

### Get the most recent media published by a user.

Get the most recent media published by a user. To get the most recent media published by the owner of the access token, you can use `self` instead of the `user_id`.

Security scope `public_content` is required to read information about other users.

## REQUEST

### PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
<code>*user_id</code>	string		The ID of a user to get recent media of, or <code>**self**</code> to retrieve media of authenticated user.

### QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
<code>count</code>	integer		Count of media to return.
<code>max_timestamp</code>	int64		Return media before this UNIX timestamp.
<code>min_timestamp</code>	int64		Return media after this UNIX timestamp.
<code>min_id</code>	string		Return media later than this <code>`min_id`</code> .
<code>max_id</code>	string		Return media earlier than this <code>`max_id`</code> .

## RESPONSE

STATUS CODE - 200: Users media entries.

### RESPONSE MODEL - application/json

```
{  
  data [{  
    Array of object:  
    attribution string ??? Unknown ???  
    caption {  
      created_time string Caption creation UNIX timestamp  
      from {  
        full_name string User full name  
        id string User ID  
        profile_picture string URL to user profile picture  
        username string User name, nickname  
      }  
      id string ID of this caption  
      text string Caption text  
    }  
    comments {  
      count integer Number of comments available, data does not necessary contain all comments  
      data [{  
        Array of object:
```

```

    created_time string Comment creation UNIX timestamp
    from {
      full_name string User full name
      id string User ID
      profile_picture string URL to user profile picture
      username string User name, nickname
    }
    id string ID of this comment
    text string Text of the comment
  ]]
}
created_time string Media creation UNIX timestamp
filter string Filter of this media entry
id string ID of a media entry
images {
  low_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  standard_resolution {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
  thumbnail {
    height integer Image/video height in pixels
    url string URL of the image/video resource
    width integer Image/video width in pixels
  }
}
likes {
  count integer Number of likes available, data does not necessary contain all comments
  data [{
    Array of object:
    full_name string User full name
    id string User ID
    profile_picture string URL to user profile picture
    username string User name, nickname
  }]
}
link string Fixed URL of this media entry
location {
  id string ID of this location (in some responses it has a type of 'integer')
  latitude number Location latitude
  longitude number Location longitude
  name string Location name
}
tags [string]
type enum ALLOWED:image, video
Type of this media entry
user {
  full_name string User full name
  id string User ID
  profile_picture string URL to user profile picture
  username string User name, nickname
}

```

**user\_has\_liked** **boolean** Indicates whether authenticated user has liked this media or not

**users\_in\_photo** [{

Array of object:

**position** {

**x** **number** X position (horizontal)

**y** **number** Y position (vertical)

}

**user** {

**full\_name** **string** User full name

**id** **string** User ID

**profile\_picture** **string** URL to user profile picture

**username** **string** User name, nickname

}

}]

**videos** {

**low\_resolution** {

**height** **integer** Image/video height in pixels

**url** **string** URL of the image/video resource

**width** **integer** Image/video width in pixels

}

**standard\_resolution** {

**height** **integer** Image/video height in pixels

**url** **string** URL of the image/video resource

**width** **integer** Image/video width in pixels

}

}

}]

**meta** {

**code** **integer** HTTP result code

}

**pagination** {

**next\_max\_id** **string** The max ID of the next page

**next\_url** **string** URL to retrieve next page of entries

}

}



