
Pixelmerge Documentation

Release 0.1

Pixelmerge

November 23, 2016

1	Application Programming Interface	3
1.1	OAuth2	3
1.2	Requests	4
1.3	Responses	4
1.4	Permissions	5
1.5	Endpoints	5
1.6	Objects	11

Contents:

Application Programming Interface

1.1 OAuth2

Pixelmerge uses OAuth2 to authorize API Clients for specific Endpoints. We implement the latest OAuth2 Draft (at the moment 31).

1.1.1 Authorization Code Grant

Pixelmerge supports the Authorization Code Grant as preferred grant:

1. Redirect the user to <https://pixelmerge.me/oauth2/connect> with the following parameters as query components encoded as application/x-www-urlencoded ¹

response_type Expected response, must be “code” for Authorization Code Grant

client_id Your applications Client ID

redirect_uri A URL to redirect after permission grant

scope List of requested [permissions](#). Must be a space-separated list of permission tokens obtained from /api/info. If omitted, the request will implicit contain the standard scopes of your Client.

state An optional state to determine the request in your callback. We (and the draft) recommend, that you are using a state

2. The user will see a dialog, which asks him to permit or deny the request. If the user permits the authorization grant, then the user will be redirected to `redirect_url` with the following additional parameters in the Query component, which is encoded as application/x-www-urlencoded ¹:

code The authorization code to obtain an access token

state If you supplied a state in step 1, then you will get it here

3. Call https://pixelmerge.me/oauth2/access_token on server side with the following parameters as query component encoded as application/x-www-urlencoded ¹:

grant_type The OAuth2 grant type, in this case “authorization_code”

code The obtained authorization code from step 2

redirect_uri The `redirect_uri` parameter from step 1

client_id Your Client ID

¹ Hors, A., Raggett, D., and I. Jacobs, “HTML 4.01 Specification”, World Wide Web Consortium Recommendation REC-html401-19991224, December 1999.

client_secret Your Client secret

4. Step 3 will respond with an body encoded as application/json ²

1.2 Requests

The Pixelmerge API receives requests only by HTTPS. A request is sent to an endpoint, which replies with data to the request. A request takes a method, a path, an optional query with parameters, and an optional application/json-encoded body. Authorization is performed by OAuth 2, access tokens must be supplied as a query parameter called “access_token”.

At the moment, the requests are not rate-limited, so you can request until we block your IP address or access token manually. This could change in near time, so you should always accept errors.

1.2.1 Generic Collection Parameters

Some API endpoints, which return a collection, accept generic collection parameters, to subset the responses. In general, any collection, which returns objects with a time, is supported.

Name	Type	Description
count	number	Maximum amount of the objects returned
since_time	number	Instruct, that the API should not return objects with a time, which is before the time specified in this parameter as UTC Unix timestamp.

All of these parameters must be supplied as query parameters.

1.3 Responses

An API endpoint returns responses, which consist of a HTTP status code and an optional application/json-encoded body.

1.3.1 Errors

If an error while performing the request occurs, the API will return a HTTP status code equal or greater than 400. Then, the API will return also a JSON object in the body with the parameter error, which describes the error. This is compatible to OAuth2, especially because OAuth2 errors are handled by the same functionality. In addition to the error field, the object can contain other error-specific fields, which describe the errors more specific.

```
{
  "error": {string},
  ...
}
```

See [Errors](#) for a full list of API errors.

² Crockford, D., “The application/json Media Type for JavaScript Object Notation (JSON)”, RFC 4627, July 2006.

1.4 Permissions

Pixelmerge maintains several permissions for its API endpoints. These permissions are also OAuth2 scopes, that means, you have to obtain them on Authorization Grant

Name	Description
get_private_scope	Access to objects which hold a scope which is private
create_post	Create new objects which are related to posts
get_services	Read service objects
add_service	Create service objects
delete_service	Delete service objects
get_views	Read view objects
create_view	Create view objects
delete_view	Delete view objects
get_settings	Read settings
set_settings	Set settings

1.5 Endpoints

Pixelmerge provides a RESTful API interface.

1.5.1 User

GET /users/<username>

Method GET

Path /users/<username>

Parameters	Name	Type	Description
	username	string	Name of the user

Authorization None

Permissions None

Returns [User object](#)

1.5.2 Post

GET /posts/private

This endpoint returns all private posts

Method GET

Path /posts/private

Parameters This endpoint accepts all generic collection parameters

Authorization [OAuth2](#)

Permissions [get_private_scope](#)

Returns [Array of post objects](#)

GET /posts/public

Method GET

Path /posts/public

Parameters This endpoint accepts all generic collection parameters

Authorization OAuth2

Permissions None

Returns Array of post objects

GET /posts/public/<username>

Method GET

Path /posts/public/<username>

Parameters	Name	Type	Description
	username	string	Name of a registered user

In addition to this, this endpoint accepts all generic collection parameters

Authorization None

Permissions None

Returns Array of post objects

GET /posts/id/<id>

Method GET

Path /posts/id/<id>

Parameters	Name	Type	Description
	id	string	ID of a post object

Authorization None if the scope of the object is “public”, otherwise OAuth2

Permissions If the scope of the object ist “private”:

- get_private_scope

Returns Post object

POST /posts

Method POST

Path /posts

Parameters This endpoint accepts a JSON encoded object with the following parameters in the request body:

```
{
  "text": {string},
  "groups": [
    {string}
  ]
}
```

Name	Type	Description
text	string	Text for the post
groups	array	List of service groups IDs, on which members the text will be posted.

Authorization [OAuth2](#)

Permissions

- create_post

Returns

- 202 if the post is acceptable
- 400 if the request is invalid
- 401 if the user hasn't the permissions

1.5.3 Medium

GET /media/private

Method GET

Path /media/private

Parameters This endpoint accepts all generic collection parameters

Authorization [OAuth2](#)

Permissions get_private_scope

Returns [Array of medium objects](#)

GET /media/public

Method GET

Path /media/public

Parameters This endpoints accepts all generic collection parameters

Authorization [OAuth2](#)

Permissions None

Returns [Array of medium objects](#)

GET /media/public/<username>

Method GET

Path /media/public/<username>

Parameters	Name	Type	Description
	username	string	Name of a registered user

In addition to this, this endpoint accepts all generic collection parameters

Authorization None

Permissions None

Returns [Array of medium objects](#)

GET /media/id/<id>

Method GET

Path /media/id/<id>

Parameters	Name	Type	Description
	id	string	ID of a medium object

Authorization None if the scope of the object is “public”, otherwise [OAuth2](#)

Permissions If the scope of the object is not “private”:

- `get_private_scope`

Returns [Medium object](#)

1.5.4 Location

GET /locations/private

Method GET

Path /locations/private

Parameters This endpoint accepts all generic collection parameters

Authorization [OAuth2](#)

Permissions `get_private_scope`

Returns [Array of location objects](#)

GET /locations/public

Method GET

Path /locations/public

Parameters This endpoint accepts all generic collection parameters

Authorization [OAuth2](#)

Permissions None

Returns [Array of location objects](#)

GET /locations/public/<username>

Method GET

Path /locations/public/<username>

Parameters	Name	Type	Description
	username	string	Name of a registered user
	count	number	Number of records to receive as query parameter
	since_time	number	Time of the earliest object to return (query)

In addition to this, this endpoint accepts all generic collection parameters.

Authorization None

Permissions None

Returns [Array of location objects](#)

GET /locations/id/<id>

Method GET

Path /locations/id/<id>

Parameters	Name	Type	Description
	id	string	ID of a location object

Authorization None if the scope of the object is “public”, otherwise [OAuth2](#)

Permissions If the scope of the object is “private”:

- `get_private_scope`

Returns [Location object](#)

1.5.5 Permissions

GET /permissions

Method GET

Path /permissions

Parameters None

Authorization [OAuth2](#)

Permissions None

Returns Array of [Permissions](#)

```
[
  {string}
]
```

DELETE /permissions/<permission>

Method DELETE

Path /permissions/<permission>

Parameters	Name	Type	Description
	permission	string	Permission , which should removed from the current access token

Authorization [OAuth2](#)

Permissions None

Returns Nothing

1.5.6 Service

GET /services

Method GET

Path /services

Parameters None

Authorization OAuth2

Permissions get_services

Returns Array of service objects

GET /services/<id>

Method GET

Path /services/<id>

Parameters	Name	Type	Description
	id	string	ID of a service object

Authorization OAuth2

Permissions get_services

Returns Array of service objects

DELETE /services/<id>

Method DELETE

Path /services/<id>

Parameters	Name	Type	Description
	id	string	ID of a service object

Authorization OAuth2

Permissions delete_service

Returns Nothing

POST /services

1.5.7 Service groups

GET /services/groups

POST /services/groups

GET /services/groups/<id>

DELETE /services/groups/<id>

POST /services/groups/<id>/<service_id>

DELETE /services/groups/<id>/<service_id>

1.5.8 Settings

GET /settings

POST /settings

1.5.9 View

GET /views

POST /views

GET /views/<id>

POST /views/<id>

DELETE /views/<id>

1.5.10 Info

GET /info

1.6 Objects

The Pixelmerge API responds with API objects, which are specified in this section:

1.6.1 Post Objects

Post objects represent raw posts, which are not merged and imported direct from services. They contain medium and location objects.

```
{
  "id": {string},
  "external_url": {string},
  "url": {string},
  "scope": {string},
  "username": {string},
  "service": {string},

  "text": {string},
  "location": {
    "longitude": {number},
    "latitude": {number}
  },

  "media": [
    {
      "type": {string},
      "url": {string}
    }
  ],

  "time": {number},

  "creator": {
    "username": {string},
    "display_name": {string},
    "url": {string},
    "avatar": {string}
  }
}
```


Field	Type	Description
id	string	An unique ID which specifies a post object
external_url	string	A URL which references the specific object on service side.
url	string	A URL which references the specific object on Pixelmerges side
scope	string	Scope of the post object
username	string	Name which references the user object of the service owner
text	string	The text part of the post as unformatted plain text
location	object	Specifies the location of the object; null, if the post hasn't any location data.
location[longitude]	number	Longitude of the location
location[latitude]	number	Latitude of the location
media	array	List of media supplied by the post
media[]	object	Object which holds the media data
media[][type]	string	Type of the media, at the moment the following types are supported: <ul style="list-style-type: none"> image
media[][url]	string	URL which references the source of the medium
time	number	Unix Timestamp (UTC) which specifies the creation datetime of the post
creator	object	Object which holds information of the creator on service side
creator[username]	string	Username of the creator of the post on service side
creator[display_name]	string	Display name of the creator of the post on service side
creator[url]	string	Link to the resource of the creator on service side
creator[avatar]	string	Link to the user image on service side (Avatar)

1.6.2 User Objects

User objects specify Pixelmerge users.

```
{
  "username": {string},
  "created_at": {number},
  "views": [
    {string}
  ]
}
```

Field	Type	Description
username	string	Name of the user
created_at	number	Time of the creation of the user
views	array	List of public user views
views[]	string	Name of a user view

1.6.3 Location Objects

Location objects represent locations from posts, but extracted to query them directly.

```
{
  "id": {string},
  "username": {string},
  "service": {string},
  "time": {number},
  "position": {
    "longitude": {number},
    "latitude": {number}
  },
  "scope": {string},
  "post": {string}
}
```

Field	Type	Description
id	string	Unique identifier of the location
username	string	Name of the location object owner
service	string	ID which references the service object associated with the location object
time	number	Time of the creation of the post related to this location object
position	object	Object which holds the location information
position[longitude]	number	Longitude of the position
position[latitude]	number	Latitude of the position
scope	string	Scope of the location object
post	string	ID which references the post object, from which the location data was obtained.

1.6.4 Medium Objects

Medium objects represent any type of mediums, e.g. images, videos, documents.

```
{
  "id": {string},
  "username": {string},
  "service": {string},
  "time": {number},
  "url": {string},
  "scope": {string},
  "post": {string},
  "type": {string}
}
```

Field	Type	Description
id	string	An unique ID which specifies a media object
username	string	Name which references a user which owns the post
service	string	An ID which references the service object
time	number	Time of the creation of the referenced post object
url	string	URL which references the source of the medium
scope	string	Scope of the medium
post	string	ID which references the post object
type	string	Type of the medium

1.6.5 Service Objects

Service objects represent external services from which Pixelmerge imports posts.

```
{
  "id": {string},

  "username": {string},
  "service": {string},
  "state": {string},
  "realname": {string}
}
```

Field	Type	Description
id	string	An unique ID which specifies a service object
username	string	Name which references the user object of the owner of the service
service	string	Name of the external service (e.g. twitter)
state	string	State of the service. Valid states are “pending”, “inactive”, “active”
realname	string	Display name of the service account

1.6.6 View Objects

View objects represent the meta data used to process posts.

```
{
  "id": {string},

  "settings": {object},
  "order": {number},
  "type": {string},

  "scope": {string}
}
```

Field	Type	Description
id	string	An unique ID which specifies a view object
settings	object	Type-specific settings for the view
order	number	Number by which the service is sorted in the user interface
type	string	Type of the view
scope	string	Scope of the view