Posts

Posts are the main section of e621. They are media uploaded by users or the artists themselves. They contain a set of *tags* that describe the post, *notes* about the post. They can also be *commented* on. A lot of e621 'actions' revolve around them.

An index of all most recent posts can be found here (https://e621.net/post/index).

GET Endpoints

Check MD5

Check the md5 hash of a post

Base URL is `https://e621.net/post/check_md5.json`

Parameters

- **md5** - The md5 hash of the file to match

Typical Response Example

```
{
    md5: string
    exists: boolean
    post_id: number
}
```

Example JSON Request + Response (https://e621.net/post/check_md5.json?md5=e9fbd2f2d0703a9775f245d55b9a0f9f)

Example XML Request + Response (https://e621.net/post/check_md5.xml?md5=e9fbd2f2d0703a9775f245d55b9a0f9f)

Deleted Index

Get the index of deleted posts

Base URL is `https://e621.net/post/deleted_index.json`

Note: the following fields are not available for deleted posts: `source`, `sources`, `md5`, `file_size`, `file_ext`, `preview_width`, `preview_height`, `sample_url`, `sample_width`, `sample_height`, `has_children`, `children`

Parameters

- **user_id** - ID of the user to filter post uploads by
- **page** - The page number to return

Typical Response Example

```
[
    {
        id: number,
        tags: string,
        locked_tags: string,
        description: string,
        created_at: {
            json_class: string,
            s: number,
            n: number
        },
        creator_id: number,
        author: string,
        change: number,
        score: number,
        fav_count: number,
        file_url: string,
        rating: string,
        status: string,
        width: number,
        height: number,
        has_comments: boolean,
        has_notes: boolean,
        parent_id: number,
        artist: string[],
        delreason: string
    },
    // ...
]
```

[Example JSON Request + Response (https://e621.net/post/deleted_index.json)](https://e621.net/post/deleted_index.json)

[Example XML Request + Response (https://e621.net/post/deleted_index.xml)](https://e621.net/post/deleted_index.xml)

List

List all posts, filtered by a set of given parameters.

The most efficient method to iterate a large number of posts is to use `before_id` starting at the highest ID, and then successively request the lowest ID returned each time. When iterating using page, posts will shift between pages if posts meeting the tags search criteria are deleted or added to the site between requests

Note: Page numbers greater than 750 will return an error.

Note: Deleted posts are not returned.

Note: Deleted posts still count towards the number of posts returned in `page` requests if `status:any` or `status:delete` is part of the tags search criteria. Deleted posts are not returned either way. JSON requests do not report any post count.

Note: Sorting results with `order:` using tags does nothing in conjunction with `before_id`.

Base URL is `https://e621.net/post/index.json`

Parameters

- **limit** - Number of posts you want to retrieve per page. There is a hard limit of 320 posts per request. Defaults to the value set in user preferences (if any)
- **before_id** - Returns the next `limit` posts with IDs lower than the given ID
- **page** - The page number to return
- **tags** - The tag search query, only returns posts that match the given spcae-delimited tags
- **typed_tags** - Set to `true` to return typed tag information. The tags value returned is a dictionary with each tag type as a key and then a list of tags of that type

Typical Response Example

```
[
    {
        id: number,
        tags: string,
        locked_tags: string,
        description: string,
        created_at: {
            json_class: string,
            s: number,
            n: number
        },
        creator_id: number,
        author: string,
        change: number,
        source: string,
        score: number,
        fav_count: number,
        md5: string,
        file_size: number,
        file_url: string,
        file_ext: string,
        preview_url: string,
        preview_width: number,
        preview_height: number,
        sample_url: string,
        sample_width: number,
        sample_height: number,
        rating: string,
        status: string,
        width: number,
        height: number,
        has_comments: boolean,
        has_notes: boolean,
        has_children: boolean,
        children: number,
        parent_id: number,
        artist: string[],
        sources: string[],
        delreason: string
    },
    // ...
]
```

Example JSON Request + Response (https://e621.net/post/index.json?tags=fluffy+rating:s&limit=5)

Example XML Request + Response (https://e621.net/post/index.xml?tags=fluffy+rating:s&limit=5)

Popular

Get popluar posts. The popularity interval is based off of the base URL.

Base URLs are `/post/popular_by_day.json`, `/post/popular_by_week.json`, and `/post/popular_by_month.json`

Parameters

NONE

Typical Response Example

```
[
    {
        id: number,
        tags: string,
        locked_tags: string,
        description: string,
        created_at: {
            json_class: string,
            s: number,
            n: number
        },
        creator_id: number,
        author: string,
        change: number,
        source: string,
        score: number,
        fav_count: number,
        md5: string,
        file_size: number,
        file_url: string,
        file_ext: string,
        preview_url: string,
        preview_width: number,
        preview_height: number,
        sample_url: string,
        sample_width: number,
        sample_height: number,
        rating: string,
        status: string,
        width: number,
        height: number,
        has_comments: boolean,
        has_notes: boolean,
        has_children: boolean,
        children: number,
        parent_id: number,
        artist: string[],
        sources: string[],
        delreason: string
    },
    // ...
]
```

Example JSON Request + Response (https://e621.net/post/popular_by_day.json)

Example XML Request + Response (https://e621.net/post/popular_by_day.xml)

Show

Show a single post's information by ID or MD5 hash

Base URL is `https://e621.net/post/show.json`

Note: Do not use the /post/show API for bulk requests. Use of /post/index allows returning multiple results and is more efficient, while also returning the same information.

Parameters

- **id** - ID of the post to retrieve
- **md5** - MD5 hash of the post to retrieve

Typical Response Example

```
    id: number,
    tags: string,
    locked_tags: string,
    description: string,
    created_at: {
        json_class: string,
        s: number,
        n: number
    },
    creator_id: number,
    author: string,
    change: number,
    source: string,
    score: number,
    fav_count: number,
    md5: string,
    // in bytes
    file_size: number,
    file_url: string,
    file_ext: string,
    preview_url: string,
    preview_width: number,
    preview_height: number,
    sample_url: string,
    sample_width: number,
    sample_height: number,
    rating: string,
    status: string,
    width: number,
    height: number,
    has_comments: boolean,
    has_notes: boolean,
    has_children: boolean,
    children: number,
    parent_id: number,
    artist: string[],
    sources: string[],
    delreason: string
```

Example JSON Request + Response (https://e621.net/post/show.json?id=8595)

Example XML Request + Response (https://e621.net/post/show.xml?id=8595)

Tags

Get the tags of a post by ID or MD5 hash

Base URL is `https://e621.net/post/tags.json`

Parameters

- **id** - ID of the post to retrieve the tags of
- **md5** - MD5 hash of the post to retrieve of

Typical Response Example

```
    id: number,
    name: string,
    count: number,
    // 0 General
    // 1 Artist
    // 3 Copyright
    // 4 Character
    // 5 Species
    type: e621TagTypes,
    type_locked: boolean
```

Example JSON Request + Response (https://e621.net/post/tags.json?id=8595)

Example XML Request + Response (https://e621.net/post/tags.xml?id=8595)

POST Endpoints

Create

Create a new post. There are only four mandatory fields: you need to supply the tags, and you need to supply the file, either through a multipart form or through a source URL. A source, even if blank, and a rating are also required

Base URL is `https://e621.net/post/create.json`

POST Parameters

`post[tags]`, `post[file]`, `post[rating]` and `post[source]` are **required**

- **post[tags]** - A space-delimited list of tags for the post
- **post[file]** - The file data encoded as a multipart form
- **post[rating]** - The rating for the post. Can be `safe`, `questionable`, or `explicit`
- **post[upload_url]** - If this is a URL, e621 will download the file
- **post[source]** - This will be used as the post's 'Source' text. Separate multiple URLs with %0A (url-encoded newline) to define multiple sources. Limit of five URLs
- **post[description]** - The description for the post
- **post[is_rating_locked]** - Set to `true` to prevent others from changing the rating
- **post[is_note_locked]** - Set to `true` to prevent others from adding notes
- **post[parent_id]** - The ID of the parent post (if any)

Example POST Object

```
{
    // file is extracted from the URL
    "post[source]": "some url",
    "post[tags]": "tag1 tag2 tag3",
    "post[rating]": "explicit",
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```

If the post upload succeeded, you'll get an attribute called `location` in the response pointing to the relative URL of your newly uploaded post.

If the call fails, the following response reasons are possible:

MD5 mismatch - This means you supplied an MD5 parameter and what e621 got doesn't match. Try uploading the file again.
duplicate - This post already exists in e621 (based on the MD5 hash). An additional attribute called location will be set, pointing to the (relative) URL of the original post.
other - Any other error will have its error message printed.

Destroy

Delete a post by ID. You must either own the post or be a moderator

Base URL is `https://e621.net/post/destroy.json`

Note: The mode parameter is only required (to be 1) if you are attempting to permanently destroy the post (which must be called a second time, after the post has been normally deleted). You must be logged in to use this action. You must also be janitor or higher

POST Parameters

`id` and `reason` are **required**

- **id** - ID of the post to delete
- **reason** - Reason the post is being deleted
- **mode** - Set to `1` if you are attempting to permanently destroy this post (will only work if called on an already deleted post)

Example POST Object

```
{
    "id": 114,
    "reason": "duplicate"
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```

Flag

Flaqg a post for deletion

Base URL is `https://e621.net/post/flag.json`

POST Parameters

`id` and `flag_option` are **required**

- **id** - ID of the post to flag for deletion
- **inferior_parent** - ID of the post which is superior to thepost being flagged. For duplicates, this should be the ID of the post which is older. Use only when `flag_option` is set to `inferior`
- **flag_option** - Indicates the reason the post should be deleted. Valid values are: `uploader` (Uploader requests deletion) `inferior` (Repost/inferior version of existing post) or one of the following
    - 1 Artist is on avoid-posting list
    - 2 Post is paysite material
    - 3 Uncredited trace
    - 4 Real-life pornography
    - 5 File corrupted
    - 6 Image previously deleted

Example POST Object

```
{
    "id": 113,
    "flag_option": 4
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```

Revert Tagss

Revert a post's tags to previous version

Base URL is `https://e621.net/post/revert_tags.json`

POST Parameters

Required parameters

- **id** - ID of the post to revert the tags of
- **history_id** - Version number to revert the tags to

Example POST Object

```
{
    "id": 11345,
    "history_id": 1
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```

Update

Update a post's information

Base URL is `https://e621.net/post/update.json`

POST Parameters

`id` is **required**

- **id** - ID of the post to update
- **post[tags]** - An updated set of space-delimited tags for the post
- **post[old_tags]** - The old tags from the post
- **post[rating]** - The updated rating for the post
- **post[source]** - the updated source set for the post
- **post[description]** - The description for the post
- **post[is_rating_locked]** - Set to `true` to prevent others from changing the rating
- **post[is_note_locked]** - Set to `true` to prevent others from adding notes
- **post[parent_id]** - The ID of the parent post
- **reason** - The reason for the submitted changes. Inline DText allowed

Example POST Object

```
    "id": 11456,s
    "post[rating]": "explicit",
    "post[parent_id]": 11234
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```

Vote

Vote for a post. You can vote once per post per IP address

Base URL is `https://e621.net/post/vote.json`

POST Parameters

`id` is **required**

- **id** - ID of the post to vote for

- **score** - Set to `1` to upvote, `-1` to downvote

Example POST Object

```
{
    "id": 11134,
    "score": 1
}
```

Response

Response object should look similar to the object below. `?` representing a response property that may or may not be returned

```
    success: boolean,
    reason?: string,
    message?: string
```