

# Artists

Artists are just as they seem. They represent e621 *users* or even external users from another site that create content. This information is used in any *posts* made to e621.

You can check out how e621 uses this data [here \(https://e621.net/artist/index\)](https://e621.net/artist/index).

## GET Endpoints

### List

List artists, filtering by a set of given parameters

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

#### Parameters

- **name** - Name or part of the name of the artist (or their URL)
- **limit** - Number of records per page to return
- **order** - Order of the returned response entries. Can be `date` or `name`
- **page** - The page number to return

#### Typical Response Example

```
[
  {
    id: number,
    name: string,
    other_names: string,
    group_name: string,
    urls: string[],
    is_active: boolean,
    version: number,
    updater_id: number
  },
  // ...
]
```

[Example JSON Request + Response \(https://e621.net/artist/index.json?name=bubble\)](https://e621.net/artist/index.json?name=bubble)

[Example XML Request + Response \(https://e621.net/artist/index.xml?name=bubble\)](https://e621.net/artist/index.xml?name=bubble)

## POST Endpoints

### Create

Create a single new artist

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

#### POST Parameters

`artist[name]` and `artist[urls]` are **required**

- **artist[name]** - The artist's name
- **artist[urls]** - A list of URLs where the artist can be found
- **artist[group\_name]** - The group the artist is a member of
- **artist[other\_names]** - List of comma-separated names the artist is also known as

#### Example POST Object

```
{
  "artist[name]": "foxxo",
  "artist[urls]": "URL_1 URL_2 URL_3",
  "artist[group_name]": "foxes united",
  "artist[other_names]": "super_foxxo,test,other_test"
}
```

## 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
```

## Destroy

Delete/destroy an artist by ID

Base URL is <https://e621.net/artist/destroy.json>

### POST Parameters

**id** is required

- **id** - The ID of the artist to delete/destroy

### Example POST Object

```
{
  "id": 123456
}
```

## 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 an artist's information by their ID

Base URL is <https://e621.net/artist/update.json>

### POST Parameters

Only the artist's **id** is required

- **id** - ID of the artist to update
- **artist[name]** - The artist's updated name
- **artist[urls]** - A list of URLs where the artist can be found
- **artist[is\_active]** - Whether or not the artist is active
- **artist[group\_name]** - The group the artist is a member of
- **artist[other\_names]** - List of comma-separated names the artist is also known as

### Example POST Object

```
{
  "artist[name]": "foxxo_new",
  "artist[urls]": "URL_1 URL_2 URL_3",
  "artist[group_name]": "foxes united",
  "artist[is_active]": true,
  "artist[other_names]": "super_foxxo,test,other_test,updated_name"
}
```

## 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
```