

Projects: Create, list, update, archive & delete

This guide walks you through creating, updating and archiving projects using our Open API.

Jump to section:

- [POST /projects/create](#)
 - [POST/projects/list](#)
 - [PATCH / projects/update](#)
 - [POST /projects/archive](#)
 - [POST /projects/unarchive](#)
 - [DELETE /projects/softDelete](#)
 - [DELETE /projects/hardDelete](#)
-

POST /projects/create

Use this endpoint to create new projects.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/projects/create
```

Headers:

- `accept: application/json`
- `X-API-KEY: YOUR_API_KEY` (Replace `YOUR_API_KEY` with your actual API key.)
- `Content-Type: application/json`

Request structure

Every request to create a new project must include a JSON object with the following:

Owner field: The `owner` should be a valid email address of an Easy PV Pro team member. This email will become the primary project owner.

Meta object: Use the `meta` object to pass in parameters for the project. All fields are optional.

- **projectType** Sets the design mode for the project. Allowed values:
 - "three-d-model"
 - "quick-roof"
 - "roof-outline"
- **"magicMode" : true** Enable automatic roof scanning - only available for three-d-model projects
- `ProjectName`
- `customerName`
- `customerEmail`
- `customerPhone`
- `address`
- `postcode`
- `lat` and `lng` - provide the latitude and longitude in the [decimal degrees](#) format
- `zoom` - set the zoom level of the satellite map imagery
- `status` - set the status of the project. Allowed values:
 - "Lead"
 - "Quote"
 - "Sale"
 - "Install"
 - "Completed"
 - "Rejected"
- `crmReference` - include an ID or reference to the customer record or project in your CRM

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to create a project with the required fields:

```
curl -X POST 'https://easy-pv.co.uk/api/v1/projects/create' \  
  -H 'accept: application/json' \  
  -H 'X-API-KEY: YOUR_API_KEY' \  
  -H 'Content-Type: application/json' \  
  -d '{
```

```
"owner": "sales@example.com",
"meta": {
  "projectType": "three-d-model",
  "magicMode": "true",
  "customerName": "Joe Bloggs",
  "address": "123 Sample Street",
  "postcode": "AB12 3CD"
}
```

Response

A successful call returns a JSON object containing a `projectId` which you should store in your own database for future reference.

You can use the ID to construct a link directly to the project using the following link structure:

```
https://easy-pv.co.uk/?project=[PROJECT ID]
```

POST /projects/list

Use this endpoint to get a list of projects.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/projects/list
```

Headers:

- `X-API-KEY: [YOUR_API_KEY]` (Replace `[YOUR_API_KEY]` with your actual API key.)

Request structure

Every request to update a project must contain a JSON object with the following.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the projects.

Start date: `start` should be in the format (YYYY-MM-DD)

End date: `end` should be in the format (YYYY-MM-DD)

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to retrieve a list of projects:

```
curl -X POST 'https://easy-pv.co.uk/api/v1/projects/list' \  
-H 'X-API-KEY: YOUR_API_KEY' \  
-d '{  
  "ownerEmail": "matt.agnes+proBasic@midsummerenergy.co.uk",  
  "start": "2025-05-22",  
  "end": "2025-11-22"  
}'
```

Response

A successful request returns a `projects` object containing a list of projects within the specified date range. For example:

```
{  
  "status": "success",  
  "projects": [  
    {  
      "ID": 1353680,  
      "owner": 27091,  
      "dateCreated": "2025-10-21T12:35:57.000Z",  
      "dateModified": "2025-10-21T12:35:57.000Z",  
      "projectName": "New project 10/21/2025, 1:35:57 PM",  
      "customerName": "Test Project 1",  
      "address": "",  
      "postcode": "",
```

```
    "lat": null,  
    "lng": null,  
    "status": ""  
  },  
  ...  
],  
"ownerEmail": "matt.agnes+proBasic@midsummerenergy.co.uk",  
"ownerID": 00123  
}
```

PATCH /projects/update

Use this endpoint to update core project data for an existing project.

Authentication

Endpoint:

```
PATCH https://[DOMAIN]/api/v1/projects/update
```

Headers:

- accept: application/json
- X-API-KEY: [YOUR_API_KEY] (Replace [YOUR_API_KEY] with your actual API key.)
- Content-Type: application/json

Request structure

Every request to update a project must contain the project id, user email, and new data to be updated. The `data` must be an object with keys for each field to be updated.

Project ID: The `projectId` as a number.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.

Data object: Include a `data` object that will contain the information to be updated. We currently support a `meta` object with keys for each field to be updated. See the `projects/create` and `projects/data` endpoint documentation for the list of project meta fields that can be updated.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to update a project:

```
curl -X PATCH 'https://easy-pv.co.uk/api/v1/projects/update' \  
-H 'accept: application/json' \  
-H 'X-API-KEY: YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "projectId": 986206,  
  "userEmail": "matt.agnes+pro@midsummerenergy.co.uk",  
  "data": {  
    "meta": { "status": "Completed"}  
  }  
'
```

Response

If the request is successful you will receive a 204 success response.

POST /projects/archive

Use this endpoint to archive a project.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/projects/archive
```

Headers:

- `accept: application/json`
- `X-API-KEY: [YOUR_API_KEY]` (Replace `[YOUR_API_KEY]` with your actual API key.)
- `Content-Type: application/json`

Request structure

Every request to archive a project must contain the project id and user email.

Project ID: The `projectId` as a number.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to archive a project:

```
curl -X POST 'https://easy-pv.co.uk/api/v1/projects/archive' \  
  -H 'accept: application/json' \  
  -H 'X-API-KEY: YOUR_API_KEY' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "projectId": 986206,  
    "userEmail": "matt.agnes+pro@midsummerenergy.co.uk",  
  }'
```

Response

If the request is successful you will receive a 204 success response.

POST /projects/unarchive

Use this endpoint to unarchive a project.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/projects/unarchive
```

Headers:

- `accept: application/json`
- `X-API-KEY: [YOUR_API_KEY]` (Replace `[YOUR_API_KEY]` with your actual API key.)
- `Content-Type: application/json`

Request structure

Every request to archive a project must contain the project id and user email.

Project ID: The `projectId` as a number.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to unarchive a project:

```
curl -X POST 'https://easy-pv.co.uk/api/v1/projects/unarchive' \  
  -H 'accept: application/json' \  
  -H 'X-API-KEY: YOUR_API_KEY' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "projectId": 986206,  
    "userEmail": "matt.agnes+pro@midsummerenergy.co.uk",
```

```
}'
```

Response

If the request is successful you will receive a 204 success response.

DELETE /projects/softDelete

Use this endpoint to reversibly soft-delete a project.

Authentication

Endpoint:

```
DELETE https://[DOMAIN]/api/v1/projects/softDelete
```

Headers:

- `accept: application/json`
- `X-API-KEY: [YOUR_API_KEY]` (Replace `[YOUR_API_KEY]` with your actual API key.)
- `Content-Type: application/json`

Request structure

Every request to archive a project must contain the project id and user email.

Project ID: The `projectId` as a number.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to soft delete a project:

```
curl -X DELETE 'https://easy-pv.co.uk/api/v1/projects/softDelete' \  
-H 'accept: application/json' \  
-H 'X-API-KEY: YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "projectId": 986206,  
  "userEmail": "matt.agnes+pro@midsummerenergy.co.uk",  
}'
```

Response

If the request is successful you will receive a 204 success response.

`DELETE` /projects/hardDelete

Use this endpoint to permanently delete a project, together with its associated data.

Authentication

Endpoint:

```
DELETE https://[DOMAIN]/api/v1/projects/hardDelete
```

Headers:

- `accept: application/json`
- `X-API-KEY: [YOUR_API_KEY]` (Replace `[YOUR_API_KEY]` with your actual API key.)
- `Content-Type: application/json`

Request structure

Every request to archive a project must contain the project id and user email.

Project ID: The `projectId` as a number.

User email: The `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to hard delete a project:

```
curl -X DELETE 'https://easy-pv.co.uk/api/v1/projects/hardDelete' \  
-H 'accept: application/json' \  
-H 'X-API-KEY: YOUR_API_KEY' \  
-H 'Content-Type: application/json' \  
-d '{  
  "projectId": 986206,  
  "userEmail": "matt.agnes+pro@midsummerenergy.co.uk",  
}'
```

Response

If the request is successful you will receive a 204 success response.

We have updated the attribute used by the API when specifying the email address of the user that owns the record so it is consistent across all our endpoints. We now always refer to this attribute as `userEmail`. If you previously used `ownerEmail` this will still work as it has been set up as an alias of `userEmail`.

Revision #48

Created 6 September 2024 22:16:09 by Andy

Updated 9 January 2026 10:35:58 by Saira Noor