Developer guidelines

Easy PV and Heatpunk have a number of API endpoints that can be used to integrate CRMs, job management platforms and other applications. Find out how here!

Getting started

Our API for Easy PV and Heatpunk is open, and we welcome developers working on new integrations for CRMs, job management platforms and other applications. If you have an application that you would like to connect please give us a shout - we'd be really happy to help.

Documentation and testing

Live documentation for the API can be found at https://easy-pv.co.uk/api/about/. You can alternatively view the full OpenAPI schema by navigating to `[DOMAIN]/api/openapi/`

You can test calls from the documentation itself. It will also give example CURL calls which you can test from a command line. Platforms such as Postman or Hoppscotch are also very useful for development of an API integration.

Production and sandbox environments

In all the examples we give here we use the production API at the https://easy-pv.co.uk/ domain. If you are developing an integration with another application where you feel it would be helpful to test on a pre-prod environment we can give you access to a sandbox environment. Please ask for details.

Generating an API key

Most API endpoints that we make publicly available can only be accessed through a 'pro' team API key or an enterprise API key. For 'Pro' team owners and admins please login and go to your 'Pro account settings' dashboard, then click on the 'CRM connections' tab. Click on the 'Generate key' button to create an API key.

It is very important that you keep the API key secure. Anyone who gains access to it will be able to access and change all data for the users in your team, including customer names and addresses. We recommend it is only used for server-server requests, and stored as an environment variable and kept out of git repositories. If you wish to keep a backup of the key, use a secure password manager.

Note that we don't keep a record of the key. If you lose it, you will need to generate a new one for your application.

If you are developing an application just for one company, you will only need to generate one key. If you have a platform that many companies use, every company will need to generate their own key and save it to their account on your platform.

API endpoints

Live documentation for all available API endpoints can be accessed by visiting https://easy-pv.co.uk/api/about/. You can alternatively view the full OpenAPI schema by navigating to [DOMAIN]/api/openapi/

API endpoints

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

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`.

Projects: Forms & files

Within Easy PV and Heatpunk, users complete forms and generate PDF reports. In your application you may want to view the completed files and forms associated with a project, and fetch completed form data and PDF documents.

Jump to section:

`GET` /forms/list

Use this endpoint to get a list of completed forms for a project.

Authentication

Endpoint:

```
GET https://[DOMAIN]/api/v1/projects/forms/list
```

Headers:

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

Request structure

Every request must contain the project ID and user email as GET parameters.

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.

`userEmail` is necessary as the API key is shared amongst all members of your team, but projects aren't necessarily shared with all members of the team. When we receive the request we will check if the given email has the right to access the project.

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 GET 'https://easy-pv.co.uk/api/v1/projects/forms/list?projectId=12345&userEmail=sales%2540midsummerenergy.co.uk' \
     -H 'X-API-KEY: ---KEY---'
```

Response

A successful call returns a JSON object containing a `forms` and `surveys` object, which contains the completed forms and surveys.


```
{
   "forms": [
      {
         "id": "customerProposal"
      },
      {
         "id": "enaConnect"
      },
      {
         "id": "f08_v2"
      },
      {
         "id": "g99_50kw"
      },
      {
         "id": "scaffoldingRequest"
      },
      {
```

```
        "id": "g98"
      },
      {
        "id": "g99_A1_1"
      },
      {
        "id": "g99_A1_2"
      },
      {
        "id": "letterOfConsent"
      },
      {
        "id": "install_record"
      },
      {
        "id": "g99_A3_1"
      },
      {
        "id": "g99_A3_2"
      }
    ],
    "surveys": [
      {
        "id": "survey"
      }
    ]
}
```

GET /forms/form

Use this endpoint to retrieve form fields.

Authentication

Endpoint:

```
GET https://[DOMAIN]/api/v1/projects/forms/form
```

Headers:

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

Request structure

Request must contain the project ID and user email as GET parameters. You will also need to include the ID of the form.

Form category also must be set if you are requesting survey forms.

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.

Form: The ID of the `form` (from the response above).

Form Category: The `formCategory` currently only needs to be set if requesting survey forms.

Allowed Values:

- `"forms"`
- `"surveys"`

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 GET 'https://easy-pv.co.uk/api/v1/projects/forms/form?projectId=12345&userEmail=sales%2540midsummerenergy.co.uk&form=letterOfConsent' \
    -H 'X-API-KEY: ---KEY---'
```

Response

A successful call returns a JSON object containing a `form` object, which contains the fields in the form.


```
{
   "form": {
      "details": {
         "fullName": "Testing Testington",
         "address": "90 Example Rd\nCambridge \nUK\nCB4 2QD",
         "installCompany": "Midsummer Installations",
         "installType": "PV system with battery storage",
         "customerIsCompany": "as an individual"
      }
   }
}
```

`GET` /files/list

Use this endpoint to get a list of files saved to a project.

Authentication

Endpoint:

```
GET https://[DOMAIN]/api/v1/files/list
```

Headers:

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

Request structure

Request must contain the project ID, and user email as GET parameters. You must also set the type of the return object.

ID: The `id` of the project 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.

Type: Set the `type` property to `"projects"`

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 GET 'https://easy-pv.co.uk/api/v1/files/list?id=872034&userEmail=sales%40midsummerenergy.co.uk&type=projects' \
    -H 'X-API-KEY: ---KEY---'
```

Response

A successful call returns a JSON object containing a `fileRegistry` object. Files saved within the project will be contained in the `uploads` array of the response.


```
{
  "fileRegistry": {
    "uploads": {
      "quotation": [
        {
          "name": "Customer proposal~Quotation.pdf",
          "size": 4114053,
          "created": "1970-01-01T00:00:00.000Z",
          "modified": "2024-10-15T12:41:35.000Z",
          "format": ".pdf"
        }
      ],
      "installationRecord": [
        {
          "name": "Installation record~Installation record.pdf",
          "size": 59254,
```

```
            "created": "1970-01-01T00:00:00.000Z",
            "modified": "2024-10-15T12:58:29.000Z",
            "format": ".pdf"
         }
      ],
      "letterOfConsent": [
         {
            "name": "Multiple AC inverters  letter of consent~Letter of Consent.pdf",
            "size": 2685,
            "created": "1970-01-01T00:00:00.000Z",
            "modified": "2024-10-15T12:58:00.000Z",
            "format": ".pdf"
         }
      ],
   }
}
```

`GET` /files/file

Use this endpoint to download a file.

Authentication

Endpoint:

```
GET https://[DOMAIN]/api/v1/files/file
```

Headers:

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

Request structure

Request must contain the project ID, and user email as GET parameters. You must also set the type of the return object, file category and the name of the file.

ID: The `id` of the project 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.

Type: Set the `type` property to `"projects"`

File category: Set `fileCategory` to `"uploads"`

File: The name of the `file` (from the response above).

(Optional) **Response format:** You can specify the `responseFormat` from the allowed values:

- `"raw"` file as stored (Default).
- `"b64"` JSON object with image in base64 format and metadata.
- `"meta"` Just metadata.

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 GET 'https://easy-
pv.co.uk/api/v1/files/file?type=projects&id=872034&userEmail=andy%2Bjumptech%40midsummerenergy.c
o.uk&fileCategory=uploads&file=test%20project%20%20letter%20of%20consent~Letter%20of%20Consent.
pdf&responseFormat=b64' \
    -H 'X-API-KEY: ---KEY---'
```

Response

If the request is successful you will receive a 200 success response and returns the specified file.

Projects: Get project data

Easy PV and Heatpunk generate a full bill of materials for a project, and for our UK and Ireland sites you can very quickly place an order for the components from Midsummer. So you may like to import the shopping cart into your application and display an ordering link. Users love the ease of ordering all the kit for an installation.

The cart is based on the bill of materials generated by the software for a given project. The bill of materials for the project is refreshed every time the overview page of the project is opened by a user in Easy PV or Heatpunk.

`GET` /projects/data

The main endpoint for getting information about a project. A successful call returns an object with two properties:

- A `projectData` object. This includes core project data, the `cart`, and an order link to purchase the items in the cart from Midsummer.
- A `projectComponents` object. This is a dictionary containing information about the components used in the project.

Core project data

The `projectData` object is a large object containing a variety of core project and customer information. Some of the fields are exposed for legacy reasons - we'll do our best to support everything found there, but in new code we recommend only using:

- `lat` - Latitude and longitude of the property, set from the satellite map when the project is created.
- `lng`
- `address`
- `postcode`
- `dateCreated` - A timestamp for when the project was first created.
- `dateModified` - A timestamp for when the project was last saved.
  - "Modified" is a little misleading - projects are frequently auto-saved, so this timestamp usually represents when the project was last opened.

- `customerName`
- `customerPhone`
- `customerEmail`
- `projectName`
- `projectType` - In EasyPV, this contains the method used to create the roofs in this project. Can be one of:
  - `"three-d-model"` - A 3D model of the buildings, created either by the user or via "magic mode".
  - `"roof-outline"` - The outline of the roofs is drawn over a satellite image.
  - `"quick-roof"` - The dimensions of the roofs are entered by hand.
  - `"heat"` - All Heatpunk projects have this type.
- `magicMode` - If the building was automatically modelled from satellite imagery using "magic mode".
- `status` - The current "status" of the project, set by the user. Can be one of:
  - `"Lead"` - The initial status of a project created in EasyPV.
  - `"Quote"`
  - `"Sale"`
  - `"Install"`
  - `"Completed"`
  - `"Rejected"`
  - `""` - Project statuses are only available to pro users, so if a project was created before a user upgraded to pro this field may be empty.
- `projectID` - The ID of the project. Should be identical to the one provided in order to use the endpoint.
- `cart` - A complete bill of materials for the project. Described in more detail below.
- `midsummerOrderLink` - A URL that can be used to order everything needed for a project from Midsummer.

The cart

The cart is a complete list of components used in the project. Here is an example cart containing only one item:


```
"cart": [
 {
  "name": "Eurener 440W All Black N-Type Mono (Ultra Premium) solar panel",
  "qty": 38,
  "item": 93.1,
  "line": 3537.7999999999997,
  "dbID": "5910",
  "UUID": "37cc8510-0d1b-11ef-a3f8-525400a4b42c",
  "availableFrom": {
   "midsummer": {
```

```
      "ID": "5910",
      "price": 3537.7999999999997
     }
    }
   }
]
```

Each item has the following properties :

- `name` - Name of the item.
- `qty` - Number of the item in the cart.
- `item` - Price of a single copy of the item. If multiple sellers exist, this will be for the seller the user selected.
- `line` - Price for all copies of this item.
- `dbID` - Deprecated field - redundantly stores Midsummer's ID for the item.
  - We'll do our best to support this forever, but it shouldn't be used in new code.
- `UUID` - The UUID of the item in the `projectComponents` dictionary.
  - Some smaller items (like screws/rails for the mounting system, or bits of electrical cable) aren't included in `projectComponents` and won't have a UUID.
  - This will not be present on projects last modified before the 10th Feburary, or on Heatpunk projects.
- `availableFrom` - A dictionary of suppliers who the item can be purchased from.
  - Currently this will only ever contain a single entry for `"midsummer"`, with an ID for the item and the line price.

Project components

`projectComponents` is a dictionary keyed by UUID, containing information about the major components used in the project. Some components from the cart (mostly screws/rails for the mounting system, or bits of electrical cable) will not be included here.

This is currently not available on Heatpunk - the dictionary will always be empty.

While we'll try to avoid changing the UUID for any given item, it's not currently not guaranteed. We recommend against assuming that the UUID for a particular item will remain the same between requests.

```
"projectComponents": {
 "97f4a397-5b85-11ee-9d83-525400a4b42c": {
  "componentType": "inverters",
  "uuid": "97f4a397-5b85-11ee-9d83-525400a4b42c",
  "name": "Fronius Symo Advanced 3ph 15kW (and Lite)",
  "description": "3ph 15kW Symo Advanced inverter by Fronius",
  "identifiers": {
```

```
    "enaRef": "FRONI/10440/V1/A1",
    "manufacturerPartNumber": "Symo Advanced 15.0-3-M"
   },
  }
}
```


Each item has the following properties :

- `componentType` - The category of this component. Can be:
  - `"panels"`
  - `"inverters"`
  - `"optimisers"`
  - `"batteries"`
  - `"ACIsolators"`
  - `"DCIsolators"`
  - `"meters"`
  - `"extras"` - A broad category of misc items, usually referred to as "accessories" in EasyPV.
  - `"panelGroups"` - The "brand" of panel used in this project (eg: SolFiT, Jinko).
  - `"inverterGroups"` - As above, but for inverter brands.
- `uuid`
- `name`
- `description`
- `identifiers` - A dictionary of various identifiers that this component might have.
  - `enaRef` - The ENA reference number for the item, as found in the ENA Device Database.
  - `manufacturerPartNumber` - The manufacturer's model number for the item.

Authentication

Endpoint:

```
GET https://[DOMAIN]/api/v1/projects/data
```

Headers:

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

Request structure

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

ID: The `id` of the project 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 create a project with the required fields:

```
curl -X GET 'https://easy-pv.co.uk/api/v1/projects/data?projectId=1459274&userEmail=matt.agnes%2Bpro%40midsummerenergy.co.uk' \
    -H 'accept: application/json' \
    -H 'X-API-KEY: YOUR_API_KEY'
```

Response


```
{
 "status": "success",
 "projectData": {
  "lat": 52.23838103482185,
  "lng": 0.15741585962292692,
  "zoom": 20,
  "address": "87 Cambridge Road Cambridge Cambridgeshire",
  "postcode": "CB24 6AT",
  "dateCreated": "2026-02-10 09:34:12",
  "dateModified": "2026-02-10 09:44:43",
  "projectName": "Sample Project",
  "customerName": "Testing Testington",
  "includeTerrain": true,
  "MCSSiteVisitDisclaimer": false,
  "geography": {
   "altitude": 11,
   "distance": null,
```

```
      "hillSlope": null,
      "hillZone": null,
      "terrain": null,
      "topography": null,
      "windZone": 1
    },
    "projectType": "three-d-model",
    "magicMode": true,
    "customerPhone": "0777 777 777",
    "customerEmail": "testing@testington.com",
    "userID": 22811,
    "status": "Lead",
    "mapImagery": {
      "selectedType": "orthorectified",
      "orthorectified": {
        "fileId": "QjLZ1d",
        "rotationUsed": 2.2485616,
        "radius": 48.14799916152923
      }
    },
    "projectID": "1459274",
  "cart": [
    {
      "name": "Eurener 440W All Black N-Type Mono (Ultra Premium) solar panel",
      "qty": 38,
      "item": 93.1,
      "line": 3537.7999999999997,
      "dbID": "5910",
      "UUID": "37cc8510-0d1b-11ef-a3f8-525400a4b42c",
      "availableFrom": {
        "midsummer": {
          "ID": "5910",
          "price": 3537.7999999999997
        }
      }
    },
    {
```

```
  "name": "Fronius Symo Advanced 3ph 15kW (and Lite) inverter",
  "qty": 1,
  "item": 2607.99,
  "line": 2607.99,
  "dbID": "5796",
  "UUID": "97f4a397-5b85-11ee-9d83-525400a4b42c",
  "availableFrom": {
    "midsummer": {
      "ID": "5796",
      "price": 2607.99
    }
  }
},
{
  "name": "AC isolator - IMO - 32A 4-pole",
  "qty": 2,
  "item": 17.5,
  "line": 35,
  "dbID": "4495",
  "UUID": "2eb59e82-5b85-11ee-9d83-525400a4b42c",
  "availableFrom": {
    "midsummer": {
      "ID": "4495",
      "price": 35
    }
  }
},
{
  "name": "IMO 55A DC Isolator 2-pole 1-string",
  "qty": 2,
  "item": 119.22,
  "line": 238.44,
  "dbID": "4571",
  "UUID": "87402551-5b85-11ee-9d83-525400a4b42c",
  "availableFrom": {
    "midsummer": {
      "ID": "4571",
```

```
      "price": 238.44
    }
  }
},
{
  "name": "Emlite EMP1 3ph Meter",
  "qty": 1,
  "item": 110.6,
  "line": 110.6,
  "dbID": "3507",
  "UUID": "9f2d3661-5b85-11ee-9d83-525400a4b42c",
  "availableFrom": {
    "midsummer": {
      "ID": "3507",
      "price": 110.6
    }
  }
},
{
  "name": "100m reel of 4mm2 solar cable",
  "qty": 1,
  "item": 88.2,
  "line": 88.2,
  "dbID": "377",
  "availableFrom": {
    "midsummer": {
      "ID": "377",
      "price": 88.2
    }
  }
},
{
  "name": "MC4 4mm Connector Pair",
  "qty": 10,
  "item": 3.5799999999999996,
  "line": 35.8,
  "dbID": "6395",
```

```
  "availableFrom": {
    "midsummer": {
      "ID": "6395",
      "price": 35.8
    }
  }
},
{
  "name": "Label sheet",
  "qty": 1,
  "item": 2.52,
  "line": 2.52,
  "dbID": 78,
  "UUID": "8cc0c658-5b85-11ee-9d83-525400a4b42c",
  "availableFrom": {
    "midsummer": {
      "ID": 78,
      "price": 2.52
    }
  }
},
{
  "name": "Console elongation bar - set of 2",
  "qty": 38,
  "item": 24.5,
  "line": 931,
  "dbID": "5635",
  "availableFrom": {
    "midsummer": {
      "ID": "5635",
      "price": 931
    }
  }
},
{
  "name": "Console mounting bar",
  "qty": 76,
```

```
    "item": 6.55,
    "line": 497.8,
    "dbID": "2496",
    "availableFrom": {
      "midsummer": {
        "ID": "2496",
        "price": 497.8
      }
    }
  },
  {
    "name": "Console mounting clips - pack of 4",
    "qty": 38,
    "item": 4.62,
    "line": 175.56,
    "dbID": "2495",
    "availableFrom": {
      "midsummer": {
        "ID": "2495",
        "price": 175.56
      }
    }
  },
  {
    "name": "Renusol console",
    "qty": 38,
    "item": 64.4,
    "line": 2447.2000000000003,
    "dbID": "2497",
    "availableFrom": {
      "midsummer": {
        "ID": "2497",
        "price": 2447.2000000000003
      }
    }
  }
],
```

```
  "midsummerOrderLink":
"https://midsummerwholesale.co.uk/upload/?order=eyJpdGVtcyI6eyI3OCI6eyJ0aXRsZSI6IkxhYmVsIHNoZWV0IiwicXR5IjoxLCJjYXJ0T3JkZXIiOjd9LCIzNzciOnsidGl0bGUiOiIxMDBtIHJlZWwgb2YgNG1tMiBzb2xhciBjYWJsZSIsInF0eSI6MSwiY2FydE9yZGVyIjo1fSwiMjQ5NSI6eyJ0aXRsZSI6IkNvbnNvbGUgbW91bnRpbmcgY2xpcHMgLSBwYWNrIG9mIDQiLCJxdHkiOjM4LCJjYXJ0T3JkZXIiOjEwfSwiMjQ5NiI6eyJ0aXRsZSI6IlJlbnVzb2wgY29uc29sZSIsInF0eSI6MzgsImNhcnRPcmRlciI6MTF9LCIzNTA3Ijp7InRpdGxlIjoiRW1saXRlIEVNUDEgM3BoIE1ldGVyIiwicXR5IjoxLCJjYXJ0T3JkZXIiOjR9LCI0NDk1Ijp7InRpdGxlIjoiQUMgaXNvbGF0b3IgLSBJTU8gLSAzMkEgNC1wb2xlIiwicXR5IjoyLCJjYXJ0T3JkZXIiOjJ9LCI0NTcxIjp7InRpdGxlIjoiSU1PIDU1QSBEQyBJc29sYXRvciAyLXBvbGUgMS1zdHJpbmciLCJxdHkiOjIsImNhcnRPcmRlciI6M30sIjU2MzUiOnsidGl0bGUiOiJDb25zb2xlIGVsb25nYXRpb24gYmFyIC0gc2V0IG9mIDIiLCJxdHkiOjM4LCJjYXJ0T3JkZXIiOjh9LCI1Nzk2Ijp7InRpdGxlIjoiRnJvbml1cyBTeW1vIEFkdmFuY2VkIDNwaCAxNWtXIChhbmQgTGl0ZSkgaW52ZXJ0ZXIiLCJxdHkiOjEsImNhcnRPcmRlciI6MX0sIjU5MTAiOnsidGl0bGUiOiJFdXJlbmVyIDQ0MFcgQWxsIEJsYWNrIE4tVHlwZSBNb25vICh VbHRyYSBQcmVtaXVtKSBzb2xhciBwYW5lbCIsInF0eSI6MzgsImNhcnRPcmRlciI6MH0sIjYzOTUiOnsidGl0bGUiOiJNQzQgNG1tIENvbm5lY3RvciBQYWlyIiwicXR5IjoxMCwiY2FydE9yZGVyIjo2fX0sInRyYWNlIjp7InNyYyI6ImVhc3ktcHYiLCJtb2RlIjoidWsiLCJwcm9qZWN0SUQiOiIxNDU5Mjc0In0sImRlc2lnblNvdXJjZSI6ImVhc3ktcHYifQ%3D%3D",
  "performanceVisited": true,
  "solarGenerationMCSCompatibility": true,
  "autoAllocationChanged": false,
  "consumptionMCSCompatibility": true,
  "MCSConsumption": 4000,
  "MCSOccupancy": "in half the day",
  "MCSConsumptionBool": false,
  "MCSAdditionalConsumption": 0,
  "MCSReservedCapacity": 0,
  "autoSunpaths": true,
  "shadingzone": "12",
  "importTariffID": "5f4b7ffa-51af-4722-936e-d2e1a5fe3541",
  "exportTariffID": "1bafe050-aa1b-45dd-838b-a27decd26fb8",
  "totalConsumption": 4000,
  "consumptionProfile": "domesticInHalfDay",
  "selfConsumption": 1584.262476824215,
  "tariffSettings": {
    "tariffs": []
  },
  "importMCSTariffID": "5f4b7ffa-51af-4722-936e-d2e1a5fe3541",
  "exportMCSTariffID": "1bafe050-aa1b-45dd-838b-a27decd26fb8"
```

```
  },
  "projectComponents": {
    "37cc8510-0d1b-11ef-a3f8-525400a4b42c": {
      "componentType": "panels",
      "uuid": "37cc8510-0d1b-11ef-a3f8-525400a4b42c",
      "name": "Eurener 440W All Black N-Type Mono (Ultra Premium)",
      "description": "440W panel with over 22% efficiency from Eurener",
      "identifiers": {
        "manufacturerPartNumber": "MEPV 132_ULTRA_440Wp"
      }
    },
    "a7e715d6-5b85-11ee-9d83-525400a4b42c": {
      "componentType": "panelGroups",
      "uuid": "a7e715d6-5b85-11ee-9d83-525400a4b42c",
      "name": "Eurener",
      "description": "",
      "identifiers": {}
    },
    "97f4a397-5b85-11ee-9d83-525400a4b42c": {
      "componentType": "inverters",
      "uuid": "97f4a397-5b85-11ee-9d83-525400a4b42c",
      "name": "Fronius Symo Advanced 3ph 15kW (and Lite)",
      "description": "3ph 15kW Symo Advanced inverter by Fronius",
      "identifiers": {
        "enaRef": "FRONI/10440/V1/A1",
        "manufacturerPartNumber": "Symo Advanced 15.0-3-M"
      },
      "enaRef": "FRONI/10440/V1/A1"
    },
    "92ec7ea0-5b85-11ee-9d83-525400a4b42c": {
      "componentType": "inverterGroups",
      "uuid": "92ec7ea0-5b85-11ee-9d83-525400a4b42c",
      "name": "Fronius (String)",
      "description": "Fronius make highly functional efficient, reliable and powerful grid-connected inverters that work with all standard solar modules, making them the indispensable heart of every PV system.",
      "identifiers": {}
    },
```

```
"2eb59e82-5b85-11ee-9d83-525400a4b42c": {
  "componentType": "ACisolators",
  "uuid": "2eb59e82-5b85-11ee-9d83-525400a4b42c",
  "name": "AC isolator - IMO - 32A 4-pole",
  "description": "4-pole AC enclosed switch isolator for circuits up to 32A.",
  "identifiers": {}
},
"87402551-5b85-11ee-9d83-525400a4b42c": {
  "componentType": "DCisolators",
  "uuid": "87402551-5b85-11ee-9d83-525400a4b42c",
  "name": "IMO 55A DC Isolator 2-pole 1-string",
  "description": "IMO 55A DC Isolator 2-pole 1-string",
  "identifiers": {}
},
"9f2d3661-5b85-11ee-9d83-525400a4b42c": {
  "componentType": "meters",
  "uuid": "9f2d3661-5b85-11ee-9d83-525400a4b42c",
  "name": "Emlite EMP1 3ph Meter",
  "description": "Emlite 3ph meter",
  "identifiers": {
    "manufacturerPartNumber": "EMP1"
  }
},
"8cc0c658-5b85-11ee-9d83-525400a4b42c": {
  "componentType": "extras",
  "uuid": "8cc0c658-5b85-11ee-9d83-525400a4b42c",
  "name": "Label sheet",
  "description": "Safety label set",
  "identifiers": {}
},
"3c773f11-75a2-11ef-aac6-525400a4b42c": {
  "componentType": "extras",
  "uuid": "3c773f11-75a2-11ef-aac6-525400a4b42c",
  "name": "Battery Hazard Warning Label Pack",
  "description": "Hazard warning pack of labels for lithium battery installations.",
  "identifiers": {}
},
```

```
    "f62e87bd-afed-11f0-a56e-3cecefd0aaca": {
      "componentType": "extras",
      "uuid": "f62e87bd-afed-11f0-a56e-3cecefd0aaca",
      "name": "Battery Warning Label Stickers",
      "description": "Warning labels for battery installs",
      "identifiers": {}
    },
    "752861c2-b0b8-11f0-a56e-3cecefd0aaca": {
      "componentType": "extras",
      "uuid": "752861c2-b0b8-11f0-a56e-3cecefd0aaca",
      "name": "SolShare Installation Manuals and Datasheets",
      "description": "Installation manuals and datasheet for the SolShare system",
      "identifiers": {}
    },
    "f21c9555-d673-11f0-8864-3cecefd0aaca": {
      "componentType": "extras",
      "uuid": "f21c9555-d673-11f0-8864-3cecefd0aaca",
      "name": "Battery in Installation Stickers",
      "description": "Battery in Installation stickers with space for battery location to be specified.",
      "identifiers": {}
    }
  }
}
```

Customise cart and order links

If you wish to modify the bill of materials for a project and create your own `cart` object, you can use this to generate a custom order link. If you do this, you can also add some additional parameters such as a delivery address or order reference to the order (see instructions in the next section).

In its simplest form, an order can be just an object with an 'items' property. For each item, specify the the products by providing the product ID and the quantity required.

```
const order = {
    "items": {
       "2586": {
          "qty": 99
       },
      "4258": {
          "qty": 99
      }
   }
}
```

Here is a more complex order, with a shipping address, despatch date and reference:

```
const order = {
    "items": {
       "2586": {
          "qty": 99
       },
   },
  "deliveryOption": "standard",
  "shippingAddress": {
    "postcode": "CB24 6AZ",
    "add1": "Midsummer Energy",
    "add2": "Cambridge Road Industrial Estate",
    "add3": "Milton",
    "add4": "",
    "email": "sales@midsummerenergy.co.uk",
    "phone": "01223 858414",
    "firstname": "Andy",
    "lastname": "Rankin"
  },
  "despatchDate": "2023-02-08",
  "reference": "trial order"
}
```

Once you have created the JSON object containing the products required for the project you can create an order string by following these steps:

1. Stringify the object
2. Base64 encode the result of the stringified object
3. URL encode the base64 encoded string

4. Append your string to the following URL to create your custom order link
   `https://midsummerwholesale.co.uk/upload?order=[result string]`

In JavaScript the function you need to generate an order is:

```
function createOrderLink(order) {
    return
"https://midsummerwholesale.co.uk/upload?order="+encodeURIComponent(btoa(JSON.stringify(order)))
}
```

API endpoints

Leads: Fetch, create, update & delete

This guide covers the API endpoints that handle leads.

Jump to section:

`POST` /leads/fetch

Use this endpoint to get all the leads from your leads dashboard so you can see them in your CRM.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/leads/fetch
```

Headers:

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

Request structure

Every request must include `userEmail` and `furthestDate` in the body.

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

Furthest date: `furthestDate` 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 fetch leads:

```
curl  -X POST 'https://easy-pv.co.uk/api/v1/leads/fetch' \
     -H 'x-api-key: YOUR_API_KEY_HERE'
     -d '{
          "userEmail": "saira.noor@midsummerenergy.co.uk",
          "furthestDate": "2025-09-10"
        }'
```

Response

A successful call will return a JSON array of lead objects, that looks like this:

```
{
   "rows": {
      "32123": {
      "dateCreated": "2025-03-11T11:03:29.000Z",
      "status": "new",
      "customerName": "John Doe",
      "address": "8 The Rowans, Milton, Cambridge, Cambridgeshire, CB24 6YU",
      "customerEmail": "test@test.com",
      "customerPhone": "07111111111",
      ...
      },
   ...
   }
}
```

`POST` /leads/create

Use this endpoint to insert a lead that comes from sources that aren't Speedy PV.

Authentication

Endpoint:

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

Headers:

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

Request structure

The request body can take the following parameters:

- `address` can include the postcode.
- `postcode`
- `customerEmail`
- `customerName`
- `customerPhone`
- `lat` and `lng` in decimal degrees format.

Example API request and response

Example API Request

Below is an example using `curl` that demonstrates how to save a lead with the require fields:

```
curl  -X POST 'https://easy-pv.co.uk/api/v1/leads/create' \
     -H 'x-api-key: YOUR_API_KEY_HERE' \
     -d '{
          "address": "7 Erdiston Ct, Bude, EX23 8HE",
          "customerName": "John Doe",
          "customerEmail": "john.doe@example.com",
          "customerPhone": "1234567890"
        }'
```

Response

A successful call will return a JSON object containing a `leadId`.

`POST` /leads/update

Use this endpoint to update the details of a lead.

Authentication

Endpoint:

```
POST https://[DOMAIN]/api/v1/leads/update
```

Headers:

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

Request structure

Every request must include the following in the request body:

- **Lead ID:** The `leadID` of the lead you want to update.
- **User email:** `userEmail` should be a valid email address of an Easy PV Pro team member who has access to the project.
- **Fields object:** A `fields` object containing the information to be updated. See the projects/create endpoint documentation for the list of meta fields that can be updated.

Example API request and response

Example API Request

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

```
curl  -X POST 'https://easy-pv.co.uk/api/v1/leads/update' \
     -H 'x-api-key: YOUR_API_KEY_HERE' \
     -d '{
          "leadID": "1001",
          "userEmail": "saira.noor@midsummerenergy.co.uk",
          "fields": {
            "address": "123 main st"
          }
         }'
```

Response

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

`DELETE` /leads/delete

Use this endpoint to delete leads from your leads dashboard.

Authentication

Endpoint:

```
DELETE https://[DOMAIN]/api/v1/leads/delete
```

Headers:

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

Request structure

Every request must include the following in the request body:

- **Lead ID:** The `leadID` of the lead you want to update.

- **User email:** `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 save a lead with the require fields:

```
curl  -X DELETE 'https://easy-pv.co.uk/api/v1/leads/delete' \
     -H 'x-api-key: YOUR_API_KEY_HERE' \
     -d '{
          "leadID": "1001"
          "userEmail": "saira.noor@midsummerenergy.co.uk"
        }'
```

Response

If the request is successful you will receive a 200 success response.