

Developer guidelines

Heatpunk has a number of API endpoints that can be used to integrate CRMs, job management platforms and other applications. Find out how [here](#)!

For Easy PV integrations, you can find full documentation on the [Easy PV Developer Guidelines](#).

- [Getting started](#)
- [API endpoints](#)
 - [Projects: Create, list & update](#)
 - [Projects: Forms & files](#)
 - [Projects: Get project data](#)
 - [Leads: Fetch, create, update & delete](#)

Getting started

Our API for 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://heatpunk.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://heatpunk.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.

Still need help? / Want to know more?

Further guides: Take a look at our [Developer help guides](#)

Contact Support: If you've got further questions, get in touch with our support team at **help@heatpunk.co.uk** (UK) or **help@heatpunk.ie** (IE). *If you are experiencing a bug, to help us look into your issue faster, please include the email address of your account, the name of the project and screenshots of any unexpected behaviour.*

API endpoints

Projects: Create, list & update

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/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 a Heatpunk 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.

- `buildDate` Sets the design mode for the project. Allowed values:

- "pre2000"
- "post2000"
- "post2006"
- projectName
- projectType - All Heatpunk projects have the type "heat"
- customerName
- customerEmail
- customerPhone
- address
- postcode
- altitude
- lat and lng - provide the latitude and longitude in the [decimal degrees](#) format
- 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://heatpunk.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": {
      "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://heatpunk.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 a Heatpunk 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://heatpunk.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": "2025-09-01T09:14:30.000Z",
      "dateModified": "2025-12-04T11:52:18.000Z",
      "projectName": "Mon Sep 01 2025",
      "customerName": "",
      "address": "105 Sample Road Cambridge Cambridgeshire",
      "postcode": "CB1 3QD",
      "lat": 52.193912506103516,
      "lng": 0.15014299750328064,
      "status": "Lead"
    },
    ...
  ],
  "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 a Heatpunk 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://heatpunk.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.

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`.

Still need help? / Want to know more?

Further guides: Take a look at our [Developer help guides](#)

Contact Support: If you've got further questions, get in touch with our support team at **help@heatpunk.co.uk** (UK) or **help@heatpunk.ie** (IE). *If you are experiencing a bug, to help us look into your issue faster, please include the email address of your account, the name of the project and screenshots of any unexpected behaviour.*

Projects: Forms & files

Within 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](#)
- [GET /forms/form](#)
- [GET /files/list](#)
- [GET /files/file](#)

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 a Heatpunk 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://heatpunk.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": "proposal"
    },
    {
      "id": "enaConnect"
    }
  ],
  "surveys": []
}
```

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 a Heatpunk 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://heatpunk.co.uk/api/v1/projects/forms/form?projectId=12345&userEmail=sales%2540midsummerene  
rgy.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": {  
      "coverLetterHeading": "...",  
      "coverLetter": "..."  
    },  
    "quote": {  
      "quoteReference": "32836",  
      "validFor": "30 days"  
    },  
    "scope": {  
      "show": "No",  
      "scopeOfWorks": ""  
    },  
    "terms": {  
      "show": "No",  
      "terms": ""  
    }  
  }  
}
```

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 a Heatpunk 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://heatpunk.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": {
      "heatPumpCommissioningChecklist": [
        {
          "name": "Fri Dec 27 2024 -hp-commissioning~Heat Pump Commissioning Checklist.pdf",
          "size": 83175,
          "created": "1970-01-01T00:00:00.000Z",
          "modified": "2026-01-13T17:01:29.107Z",
          "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 a Heatpunk 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://heatpunk.co.uk/api/v1/files/file?type=projects&id=872034&userEmail=andy%2Bjumptech%40midsu
mmerenergy.co.uk&fileCategory=uploads&file=test%20project%20%20letter%20of%20consent~Letter%20
of%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.

Still need help? / Want to know more?

Further guides: Take a look at our [Developer help guides](#)

Contact Support: If you've got further questions, get in touch with our support team at **help@heatpunk.co.uk** (UK) or **help@heatpunk.ie** (IE). *If you are experiencing a bug, to help us look into your issue faster, please include the email address of your account, the name of the project and screenshots of any unexpected behaviour.*

Projects: Get project data

Heatpunk generates 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 Heatpunk.

GET /projects/data

The main endpoint for getting information about a project. A successful call returns an object with the following 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.
- A `billOfMaterials` object. This object contains the full BOM in the project, categorised by component type.

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`
- `altitude`
- `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` - All Heatpunk projects have the type `"heat"` .
- `status` - The current "status" of the project, set by the user. Can be one of:
 - `"Lead"` - The initial status of a project created in Heatpunk.
 - `"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": "Samsung 5kw Monobloc Heat Pump",
    "qty": 1,
    "item": 1970.64,
    "line": 1970.64,
    "dbID": "3675#4427",
    "UUID": "a6406c7a-2bee-11ef-9a17-525400a4b42c",
    "availableFrom": {
      "midsummer": {
        "ID": "3675#4427",
        "price": 1970.64
      }
    }
  }
]
```

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 aren't included in `projectComponents` and won't have a UUID.
 - This will not be present on projects last modified before the 10th February.
- `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 may not be included here. Note this will not be present for projects last modified before 12th May.

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": {
  "a6406c7a-2bee-11ef-9a17-525400a4b42c": {
    "componentType": "heatPumps",
    "uuid": "a6406c7a-2bee-11ef-9a17-525400a4b42c",
    "name": "Samsung 5kw Monobloc Heat Pump",
    "description": "Samsung heat pumps have a great track record of solid performance in the UK, the Samsung 5kW EHS Monobloc Heat Pump is an affordable, reliable, and efficient monobloc heat pump using R32 as the refrigerant, which helps lower the impact on global warming.",
    "identifiers": {
      "enaRef": "SAMSG/09641/V1",
      "manufacturerPartNumber": "AE050RXYDEG/EU"
    }
  },
}
```

Each item has the following properties :

- `componentType` - The category of this component. On Heatpunk, this can be:
 - `"heatPumps"`
 - `"hotWaterStorages"`
 - `"heatAccessories"`
 - `"heatPumpGroups"` - The "brand" of heat pump used in this project (eg: Samsung, Grant).
 - `"hotWaterStorageGroups"` - As above but for hot water storage 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.

Bill of materials

`billOfMaterials` is complete list of components used in the project, similarly to the cart in the core project data, except the `billOfMaterials` structure is **categorised by component type**, and **contains no pricing information**.

- The component types on Heatpunk are `heatPumps`, `hotWaterStorages`, and `heatAccessories`.

Each item in each category has the following information:

- `name` - Name of the item.
- `qty` - Number of this item that exist in the project.
- `itemType` - The component type (`heatPumps`, `hotWaterStorages` or `heatAccessories`)
- `uuid` - If available, the UUID of the item, which can be cross-referenced in the project components dictionary.

Authentication

Endpoint:

GET [https://\[DOMAIN\]/api/v1/projects/data](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 a Heatpunk 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://heatpunk.co.uk/api/v1/projects/data?projectId=96206&userEmail=matt.agnes%2Bpro%40midsumm
erenergy.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": "2026-05-07 15:53:23",
    "dateModified": "2026-07-06 11:58:03",
    "projectName": "Sample Project",
    "customerName": "Testing Testington",
    "buildDate": "pre2000",
    "customerPhone": "1234567890",
    "customerEmail": "email@example.com",
```

```
"projectType": "heat",
"userID": 11828,
"status": "Lead",
"ventilationParameters": {
  "assessmentMethod": "measured",
  "measuredAirPermeability": 10,
  "buildingShelterLevel": "normal"
},
"projectID": "96206",
"altitude": 122.9400024414062,
"geography": {
  "altitude": null,
  "distance": null,
  "hillSlope": null,
  "hillZone": null,
  "terrain": null,
  "topography": null,
  "windZone": null
},
"cart": [
  {
    "name": "Samsung 5kw Monobloc Heat Pump",
    "qty": 1,
    "item": 1970.64,
    "line": 1970.64,
    "dbID": "3675#4427",
    "UUID": "a6406c7a-2bee-11ef-9a17-525400a4b42c",
    "availableFrom": {
      "midsummer": {
        "ID": "3675#4427",
        "price": 1970.64
      }
    }
  },
  {
    "name": "UK Cylinders Flowcyl Slimline 150L Heat Pump Cylinder",
    "qty": 1,
    "item": 1193.58,
    "line": 1193.58,
```

```
"dbID": "5681",
"UUID": "4b60d2e0-4d93-11ef-9a17-525400a4b42c",
"availableFrom": {
  "midsummer": {
    "ID": "5681",
    "price": 1193.58
  }
},
{
  "name": "Altecnic flush and fill valve - 28mm compression",
  "qty": 1,
  "item": 68.6,
  "line": 68.6,
  "dbID": "3819",
  "availableFrom": {
    "midsummer": {
      "ID": "3819",
      "price": 68.6
    }
  },
{
  "name": "CY Cable 2-core 1mm² (10m reel)",
  "qty": 1,
  "item": 21,
  "line": 21,
  "dbID": "3821#4653",
  "availableFrom": {
    "midsummer": {
      "ID": "3821#4653",
      "price": 21
    }
  },
{
  "name": "ESBE MBA132 3-Way Motorised Ball Valve DN25",
  "qty": 1,
  "item": 119,
```

```
"line": 119,
"dbID": "7902",
"availableFrom": {
  "midsummer": {
    "ID": "7902",
    "price": 119
  }
},
{
  "name": "FastenTherm Heat Pump Foot and Strut Single",
  "qty": 2,
  "item": 16.38,
  "line": 32.76,
  "dbID": "6551",
  "availableFrom": {
    "midsummer": {
      "ID": "6551",
      "price": 32.76
    }
  },
{
  "name": "Full bore pump valve pair - 28mm x 1 1/2",
  "qty": 2,
  "item": 30.8,
  "line": 61.6,
  "dbID": "4327",
  "availableFrom": {
    "midsummer": {
      "ID": "4327",
      "price": 61.6
    }
  },
{
  "name": "Grundfos UPM4",
  "qty": 1,
  "item": 142.8,
```

```
"line": 142.8,
"dbID": "5565",
"availableFrom": {
  "midsummer": {
    "ID": "5565",
    "price": 142.8
  }
},
{
  "name": "Heat Pump Isolator - Scame - Grey - 32A 3-pole",
  "qty": 1,
  "item": 34.3,
  "line": 34.3,
  "dbID": "4562",
  "availableFrom": {
    "midsummer": {
      "ID": "4562",
      "price": 34.3
    }
  },
{
  "name": "Inta 28mm Full Bore Compression Ball Valve with Butterfly Handle",
  "qty": 2,
  "item": 15.33,
  "line": 30.66,
  "dbID": "5484",
  "availableFrom": {
    "midsummer": {
      "ID": "5484",
      "price": 30.66
    }
  },
{
  "name": "Intafil Slimline - compact 18L sealed system kit",
  "qty": 1,
  "item": 105,
```

```
"line": 105,
"dbID": "4393",
"availableFrom": {
  "midsummer": {
    "ID": "4393",
    "price": 105
  }
},
{
  "name": "Intaflex Hose Kit 1\" F x 28mm straight pair",
  "qty": 1,
  "item": 102.9,
  "line": 102.9,
  "dbID": "4784",
  "availableFrom": {
    "midsummer": {
      "ID": "4784",
      "price": 102.9
    }
  },
{
  "name": "Kamco SystemSafe&nbsp;KS8 Central Heating Hyperflush 500ml",
  "qty": 1,
  "item": 19.53,
  "line": 19.53,
  "dbID": "7568",
  "availableFrom": {
    "midsummer": {
      "ID": "7568",
      "price": 19.53
    }
  },
{
  "name": "Kamco&nbsp;SystemSafe&nbsp;KS7 Central Heating Biocide 500ml",
  "qty": 1,
  "item": 20.3,
```

```
"line": 20.3,
"dbID": "7365",
"availableFrom": {
  "midsummer": {
    "ID": "7365",
    "price": 20.3
  }
},
{
  "name": "Primary Pro 28mm x 19mm Insulation - 1 metre length (1 Metre Length)",
  "qty": 10,
  "item": 11.13,
  "line": 111.30000000000001,
  "dbID": "4741#5029",
  "availableFrom": {
    "midsummer": {
      "ID": "4741#5029",
      "price": 111.30000000000001
    }
  },
{
  "name": "Primary Pro Bond and Seal (1 unit)",
  "qty": 2,
  "item": 14.35,
  "line": 28.7,
  "dbID": "4739#5035",
  "availableFrom": {
    "midsummer": {
      "ID": "4739#5035",
      "price": 28.7
    }
  },
{
  "name": "Primary Pro Male M10 BZP Back Plate (1 unit)",
  "qty": 5,
  "item": 1.54,
```

```
"line": 7.7,
"dbID": "4746#5037",
"availableFrom": {
  "midsummer": {
    "ID": "4746#5037",
    "price": 7.7
  }
},
{
  "name": "Primary Pro Rubberlined clip for 22mm/28mm insulation (1 unit)",
  "qty": 5,
  "item": 1.7399999999999998,
  "line": 8.7,
  "dbID": "4744#5039",
  "availableFrom": {
    "midsummer": {
      "ID": "4744#5039",
      "price": 8.7
    }
  },
{
  "name": "Samsung AI Home Mono Control Kit (MIM-E03FN)",
  "qty": 1,
  "item": 642.39,
  "line": 642.39,
  "dbID": "6766",
  "availableFrom": {
    "midsummer": {
      "ID": "6766",
      "price": 642.39
    }
  },
{
  "name": "Thermox DTX heat transfer liquid (Glycol) 25L",
  "qty": 1,
  "item": 98,
```

```
"line": 98,
"dbID": "3731",
"availableFrom": {
  "midsummer": {
    "ID": "3731",
    "price": 98
  }
},
{
  "name": "Wilo Yonos PICO 25/1-8 - 180 long",
  "qty": 1,
  "item": 165.13,
  "line": 165.13,
  "dbID": "3871",
  "availableFrom": {
    "midsummer": {
      "ID": "3871",
      "price": 165.13
    }
  },
{
  "name": "World Heat Stainless Steel 25 litre wall mounted buffer (Box including brackets)",
  "qty": 1,
  "item": 455.35,
  "line": 455.35,
  "dbID": "4149",
  "availableFrom": {
    "midsummer": {
      "ID": "4149",
      "price": 455.35
    }
  },
{
  "name": "intaKlean-heatpump combined magnetic filter and strainer",
  "qty": 1,
  "item": 128.8,
```

```
"line": 128.8,
"dbID": "4392",
"availableFrom": {
  "midsummer": {
    "ID": "4392",
    "price": 128.8
  }
}
},
"midsummerOrderLink":
```

"https://midsummerwholesale.co.uk/upload/?order=eyJpdGVtcyl6eylZnZMxlp7InRpdGxlljoiVGhlcm1veCBEVFggaGVhdCB0cmFuc2ZlciBsaXF1aWQgKEdseWNvbCkgMjVMliwicXR5ljoXLCJjYXJ0T3JkZXliOjE5fSwiMzgxOSI6eyJ0aXR5ZSI6IkFsdGVjbmljIGZsdXNoIGFuZCBmaWxsIHZhbHZlIC0gMjhtbSBjb21wcmVzc2lvbilsInF0eSI6MSwiY2FydE9yZGVyIjoyfSwiMzg3MSI6eyJ0aXR5ZSI6IldpbG8gWW9ub3MgUEIDTyAyNS8xLTggLSAxODAgbG9uZylsInF0eSI6MSwiY2FydE9yZGVyIjoyMH0sljQxNDkiOmsidGl0bGUiOiJXb3JsZCBIZWF0IFN0YWlubGVzcyBTdGVlbCAyNSBsaXRyZSB3YWxsIG1vdW50ZWQgYnVmZmVyKEJveCBpbmNsdWRpbmcgYnJhY2tldHMpliwicXR5ljoXLCJjYXJ0T3JkZXliOjlfSwiNDMyNyl6eyJ0aXR5ZSI6IkZ1bGwgYm9yZSBwdW1wIHZhbHZlIHhhaXlglSAyOG1tIHggMSAxLzliLCJxdHkiOjlsImNhcnRlcmRlci6Nn0sljQzOTliOmsidGl0bGUiOiJpbmRhS2xIYW4taGVhdHB1bXAgY29tYmluZWQgbWFnbmV0aWMgZmlsdGVyIGFuZCBzdHJhaW5lcilslInF0eSI6MSwiY2FydE9yZGVyIjoyMn0sljQzOTMiOmsidGl0bGUiOiJbnRhZmlsIFNsaW1saW5lIC0gY29tcGFjdCAxOEwgc2VhbGVkIH5c3RlbnSBraXQlLCJxdHkiOjEsImNhcnRlcmRlci6MTB9LCI0NTYyljp7InRpdGxlljoiSGVhdCBQdW1wIElzb2xhdG9yIC0gU2NhbwUgLSBHcmV5IC0gMzJlBIDMtcG9sZSIsInF0eSI6MSwiY2FydE9yZGVyIjo4fSwiNDc4NCI6eyJ0aXR5ZSI6IkudGFmbGV4IEhvc2UgS2l0IDFcliBGIHggMjhtbSBzdHJhaWdodCBwYWlyIiwicXR5ljoXLCJjYXJ0T3JkZXliOjExfSwiNTQ4NCI6eyJ0aXR5ZSI6IkudGEgMjhtbSBGdWxsIEJvcuUgQ29tcHJlc3Npb24gQmFsbCBWYWx2ZSB3aXR0eJ1dHRlcmZseSBiYW5kbGUlLCJxdHkiOjlsImNhcnRlcmRlci6OX0sljU1NjUiOmsidGl0bGUiOiJHcnVuZGZvcyBVUE00IiwicXR5ljoXLCJjYXJ0T3JkZXliOjd9LCI1Njgxljp7InRpdGxlljoiVUsgQ3lsaW5kZXJzIEZsb3djeWwgU2xpbWxpbnUgMTUwTCBIZWF0IFB1bXAgQ3lsaW5kZXliLCJxdHkiOjEsImNhcnRlcmRlci6MX0sljY1NTEiOmsidGl0bGUiOiJGYXN0ZW5UaGVybSBIZWF0IFB1bXAgRm9vdCBhbmQgU3RydXQgU2luZ2xliwicXR5ljoXLCJjYXJ0T3JkZXliOjV9LCI2NzY2ljp7InRpdGxlljoiU2Ftc3VuZyBBSSBlb21lIE1vbm8gQ29udHJvbCBLaXQgKE1JTS1FMDNGTikiLCJxdHkiOjEsImNhcnRlcmRlci6MTh9LCI3MzY1ljp7InRpdGxlljoiS2FtY28mbmJzcDdtTeXN0ZW1TYWZlIm5ic3A7S1M3IENlbnRyYWwgSGVhdGluZyBCaW9jaWRiIDUwMG1slIiwicXR5ljoXLCJjYXJ0T3JkZXliOjEzfSwiNzU2OCi6eyJ0aXR5ZSI6IkthbWNVIFN5c3RlbnVhZmUmbmJzcDdtLUzggQ2VudHJhbCBIZWF0aW5nIEh5cGVyZmx1c2ggNTAwbWwiLCJxdHkiOjEsImNhcnRlcmRlci6MTJ9LCI3OTAyIjp7InRpdGxlljoiRVNCRSBNQkExMzlgMy1XYXkgTW90b3Jpc2VkiEjhbGwgVmFsdmUgRE4yNSIsInF0eSI6MSwiY2FydE9yZGVyIjo0fSwiMzY3NSM0NDI3ljp7InRpdGxlljoiU2Ftc3VuZyA1a3cgTW9ub2Jsb2MgSGVhdCBQdW1wliwicXR5ljoXLCJjYXJ0T3JkZXliOjB9LCIzODIxlzQ2NTMiOmsidGl0bGUiOiJDSWSDYWJsZSAyLWNvcuUgMW1tLCgXMG0gcmVlbCkiLCJxdHkiOjEsImNhcnRlcmRlci6M30sljQ3NDEjNTAyOSI6eyJ0aXR5ZSI6IiByaW1hcnkgUHJvIDI4bW0geCAxOW1tIEluc3VsYXRpb24gLSAxIG1ldHJlIGxlbmd0aCAoMSBNZXRyZSBMZW5ndGgpliwicXR5ljoXMCwiY2FydE9yZGVyIjoXNH0sljQ3MzkjNTAzNSI6eyJ0aXR5ZSI6IiByaW1hcnkgUHJvIEJvbmQgYW5kIFNlYWwgKDEgdW5pdCkiLCJxdHki

OjIslmNhcnRPcmRlcil6MTV9LCI0NzQ2IzUwMzciOnsidGI0bGUiOiJQcmItYXJ5IFBybyBNYWxlIE0xMCBCWIAgQmFjayBQbGF0ZSAoMSB1bml0KSIsInF0eSI6NSwiY2FydE9yZGVyIjoxNn0sljQ3NDQjNTAzOSI6eyJ0aXR5ZSI6IlByaW1hcnkgUHJvIFJlYmJlcmxpbmVklGNsaXAgZm9yIDlybW0vMjhtbSBpbmN1bGF0aW9uLCgxIHVuaXQpliicXR5Ijo1LCJjYXJ0T3JkZXliOiJlE3fX0sInRyYWNlIjp7InNyYyI6ImVhc3ktcHYiLCJtb2RlljoiaGVhdHB1bmstdWsiLCJwcm9qZWNO

SUQiOiI2NDI2NyJ9LCJkZXNpZ25Tb3VyY2UiOiJoZWFOcHVuayJ9",

```
"hpConsumption": {
  "heatMethod": "gas",
  "hwMethod": "boiler",
  "modelSolar": true,
  "modelBattery": true,
  "modelOptimisation": false,
  "performanceMethod": "scop",
  "heatTotal": 5894,
  "heatProfile": "farnborough",
  "modelWater": false,
  "gasTariff": 35,
  "gasEfficiency": 80,
  "solarArrays": [
    {
      "id": "1664cc02-9733-4155-afc5-2fcea018d96f",
      "pitch": 29,
      "orientation": 0,
      "size": 12000,
      "sf": 1
    }
  ],
  "batteries": [
    {
      "id": "b16c856b-0955-4682-a3ed-c2df73119e1f",
      "batteryCapacity": 10,
      "maxChargeRate": 50
    }
  ]
},
"totalConsumption": 4000,
"consumptionProfile": "domesticInAllDay",
"importTariffID": "d57203df-5185-443e-8e94-a367a260452c",
"exportTariffID": "57f33ffc-4816-413b-8c12-3f3426387c0a",
"emails": {
```

```
"sign-heatPunkProposal": {
  "sent": true,
  "lastSent": "30/6/2026 @ 13:43"
}
},
"projectComponents": {
  "a6406c7a-2bee-11ef-9a17-525400a4b42c": {
    "componentType": "heatPumps",
    "uuid": "a6406c7a-2bee-11ef-9a17-525400a4b42c",
    "name": "Samsung 5kw Monobloc Heat Pump",
    "description": "Samsung heat pumps have a great track record of solid performance in the UK, the Samsung 5kW EHS Monobloc Heat Pump is an affordable, reliable, and efficient monobloc heat pump using R32 as the refrigerant, which helps lower the impact on global warming.",
    "identifiers": {
      "enaRef": "SAMSG/09641/V1",
      "manufacturerPartNumber": "AE050RXYDEG/EU"
    }
  },
  "a674d244-2bee-11ef-9a17-525400a4b42c": {
    "componentType": "heatPumpGroups",
    "uuid": "a674d244-2bee-11ef-9a17-525400a4b42c",
    "name": "Samsung (Gen 6)",
    "description": "Efficient, cost-effective heat pumps from a world leader in technological solutions.",
    "identifiers": {}
  },
  "4b60d2e0-4d93-11ef-9a17-525400a4b42c": {
    "componentType": "hotWaterStorages",
    "uuid": "4b60d2e0-4d93-11ef-9a17-525400a4b42c",
    "name": "UK Cylinders Flowcyl Slimline 150L Heat Pump Cylinder",
    "description": "High quality stainless steel cylinders manufactured in the UK. Designed to work with a heat pump to achieve maximum performance, with slimline capacities ranging from 120L to 200L. Fully insulated with 50mm 100% CFC and HCFC free polyurethane and built with a 100% recyclable silver casing.",
    "identifiers": {
      "manufacturerPartNumber": "FCHPS3150"
    }
  },
  "9f421a71-4d8f-11ef-9a17-525400a4b42c": {
```

```
"componentType": "hotWaterStorageGroups",
"uuid": "9f421a71-4d8f-11ef-9a17-525400a4b42c",
"name": "UK Cylinders Slimline",
"description": "A UK based company building high quality stainless steel cylinders. Supplied with 50mm
of 100% CFC and HCFC polyurethane, a 100% recyclable silver casing and associated components. UK
Cylinders offers a 25 year warranty on all their cylinders and a 2 year warranty on parts.",
"identifiers": {}
},
"billOfMaterials": {
"heatPumps": [
{
"name": "Samsung 5kw Monobloc Heat Pump",
"qty": 1,
"itemType": "heatPumps",
"uuid": "a6406c7a-2bee-11ef-9a17-525400a4b42c"
}
],
"hotWaterStorages": [
{
"name": "UK Cylinders Flowcyl Slimline 150L Heat Pump Cylinder",
"qty": 1,
"itemType": "hotWaterStorages",
"uuid": "4b60d2e0-4d93-11ef-9a17-525400a4b42c"
}
],
"heatAccessories": [],
"heatPumpPacks": []
}
}
```

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)))  
}
```

Still need help? / Want to know more?

Further guides: Take a look at our [Developer help guides](#)

Contact Support: If you've got further questions, get in touch with our support team at **help@heatpunk.co.uk** (UK) or **help@heatpunk.ie** (IE). *If you are experiencing a bug, to help us look into your issue faster, please include the email address of your account, the name of the project and screenshots of any unexpected behaviour.*

Leads: Fetch, create, update & delete

This guide covers the API endpoints that handle leads.

Jump to section:

- [POST /leads/fetch](#)
 - [POST /leads/create](#)
 - [POST /leads/update](#)
 - [DELETE /leads/delete](#)
-

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 a Heatpunk 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://heatpunk.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://heatpunk.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 a Heatpunk 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://heatpunk.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 a Heatpunk 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://heatpunk.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.

Still need help? / Want to know more?

Further guides: Take a look at our [Developer help guides](#)

Contact Support: If you've got further questions, get in touch with our support team at **help@heatpunk.co.uk** (UK) or **help@heatpunk.ie** (IE). *If you are experiencing a bug, to help us look into your issue faster, please include the email address of your account, the name of the project and screenshots of any unexpected behaviour.*