

# 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

Use this endpoint to fetch project data.

### 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://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

A successful call returns a `projectData` object which will include core project data plus `cart` and `midsummerOrderLink` arrays.

The `projectData` object will include core project and customer information. For example:

```
"projectData": {
  "lat": 52.933899128353076,
  "lng": -1.22907350946869,
  "zoom": 19,
  "address": "1",
  "postcode": "NG93DY",
  "dateCreated": "2025-12-17 09:26:11",
  "dateModified": "2026-01-14 17:15:42",
  "projectName": "Sample Project",
  "customerName": "Test Testington",
  "buildDate": "pre2000",
  "customerPhone": "0123456789",
  "customerEmail": "email@example.com",
  "projectType": "heat",
  "userID": "10828",
  "projectID": "53792",
  "altitude": 67.37315368652344,
  "geography": {
    "altitude": null,
    "distance": null,
    "hillSlope": null,
    "hillZone": null,
    "terrain": null,
```

```
"topography": null,  
"windZone": null  
}
```

The `cart` array contains the full bill of materials as a JSON object. This contains details of all the products that have been specified in the project. For example:

```
"cart": [  
  {  
    "name": "Mitsubishi Ecodan 14kW",  
    "qty": 1,  
    "item": 7155.55,  
    "line": 7155.55,  
    "dbID": "4058#4734",  
    "availableFrom": {  
      "midsummer": {  
        "ID": "4058#4734",  
        "price": 7155.55  
      }  
    }  
  },  
  {  
    "name": "UK Cylinders 300L FlowCyl Standard Cylinder",  
    "qty": 1,  
    "item": 1349.78,  
    "line": 1349.78,  
    "dbID": "4666",  
    "availableFrom": {  
      "midsummer": {  
        "ID": "4666",  
        "price": 1349.78  
      }  
    }  
  }  
]
```

The `midsummerOrderLink` is a URL that can be used to order everything needed for a project from Midsummer. For example:



```
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)))  
}
```

---

Revision #7

Created 5 November 2024 17:24:09 by Matt Agnes

Updated 15 January 2026 14:13:55 by Saira Noor