

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 February, 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=eyJpdGVtcyl6eyl3OCi6eyJ0aXRzZSI6IkxhYmVsIHNoZWV0"
```



```
"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)))  
}
```

Revision #23

Created 8 September 2024 22:06:26 by Andy

Updated 10 February 2026 11:38:26 by Adam Hagan