

Get project details, bill of materials and order links

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.

To fetch project data, including the cart and order link, use the `api/v1/projects/data` endpoint. You need to pass in the project ID, and the user you are acting as. The user must have view rights to the project.

Sample request:

```
curl --location 'https://easy-pv.co.uk/api/v1/projects/data?projectId=[YOUR PROJECT ID]&userEmail=[YOUR USER EMAIL]' \
--header 'accept: application/json' \
--header 'X-API-KEY: [YOUR API KEY]'
```

What to expect in the response:

You will get back an object which includes `projectData`, `cart` and `midsummerOrderLink` arrays.

ProjectData

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

```
"projectData": {
  "lat": 52.2305742,
  "lng": 0.14036,
  "zoom": 19,
  "address": "101 Kings Hedges Rd, Cambridge , UK",
  "postcode": "CB4 2QD",
  "dateCreated": "2024-10-02 17:42:45",
  "dateModified": "2024-10-17 14:11:58",
  "projectName": "Sample Project",
```

```
"customerName": "Testing Testington",
"includeTerrain": false,
"MCSSiteVisitDisclaimer": false,
"geography": {
  "altitude": 12,
  "distance": "2to20",
  "hillSlope": null,
  "hillZone": null,
  "terrain": "country",
  "topography": "no",
  "windZone": 3,
  "snowZone": 2
},
"projectType": "three-d-model",
"magicMode": true,
"customerPhone": "0777 777 777",
"customerEmail": "testing@testington.com",
"userID": "19380",
"status": "Lead",
"mapImagery": {
  "selectedType": "orthorectified",
  "orthorectified": {
    "fileId": "rG2G25",
    "rotationUsed": 2.2618198,
    "radius": 48.137703120171004
  }
},
"projectID": "986206",
"MCSDisclaimer": true,
"MCSReservedCapacity": 50,
"MCSAdditionalConsumption": 0,
"MCSConsumption": "3500",
"MCSOccupancy": "home all day",
"totalConsumption": 3500,
"selfConsumption": 980.7616984962381,
"threeDModelSettingsViewDirection": [
  -4.3836904133806645,
  -6.107308078177546,
  10.644089349920092,
  0.054191298862051804,
```

```
2.2301402617071955,  
2.241462631675495  
],  
"threeDSnapshots": [  
  "l6jskm"  
],  
}
```

Cart

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": "Jinko Tiger Neo 440W N-Type All Black Mono solar panel",  
    "qty": 6,  
    "item": 84,  
    "line": 504,  
    "dbID": "5265",  
    "availableFrom": {  
      "midsummer": {  
        "ID": "5265",  
        "price": 504  
      }  
    }  
  },  
  {  
    "name": "Growatt MIN 10000 TL-X2 Three MPPT 1ph inverter",  
    "qty": 1,  
    "item": 1050,  
    "line": 1050,  
    "dbID": "5397",  
    "availableFrom": {  
      "midsummer": {  
        "ID": "5397",  
        "price": 1050  
      }  
    }  
  },  
  {
```

```

    "name": "Giv.AC 3.0 inverter",
    "qty": 1,
    "item": 933.8,
    "line": 933.8,
    "dbID": "3040",
    "availableFrom": {
      "midsummer": {
        "ID": "3040",
        "price": 933.8
      }
    }
  },
  {
    "name": "GivEnergy 5.2kWh LiFePO4 Battery",
    "qty": 1,
    "item": 1796.2,
    "line": 1796.2,
    "dbID": "4284",
    "availableFrom": {
      "midsummer": {
        "ID": "4284",
        "price": 1796.2
      }
    }
  }
],
[

```

midsummerOrderLink

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

"midsummerOrderLink":
"https://midsummerwholesale.co.uk/upload/?order=eyJpdGVtcyl6eyl3OCi6eyJ0aXRzZSI6IkxhYmVsIHNhZlV0IiwicXR5IjoyfSwiMzgwIj7InRpdGxlljoiMTAwbSByZWVzIG9mIDZtbTlgc29sYXlgyY2FibGUiLCJxdHkiOjF9LCIxNzc5Ij7InRpdGxlljoiTUM0IDZtbSBDb25uZWNOb3IglUGFpciIsInF0eSI6NH0sljlyOTgiOnsidG10bGUiOiJuaWdvIFJldHJvZm10IEZyYW1lIE1vdW50ZWQgT3B0aW1pc2VyIFRTNC1BLU8iLCJxdHkiOjZ9LClYNTg2Ij7InRpdGxlljoiRmFzdGVuc29sIHJhaWwgZ3BsaWNlliwicXR5Ijo2fSwiMjU4Nyl6eyJ0aXRzZSI6IkZhc3RlbnNvbCBByYWlslGJvbHQiLCJxdHkiOjZ9LClYNTk0Ij7InRpdGxlljoiRmFzdGVuc29sIHBvcnRyYW10IGZzYXQgdGlsZSByb29mIGhvb2siLCJxdHkiOjI0fSwiMjgxMl6eyJ0aXRzZSI6IkZhdHRIcnkgSGF6YXJkIFdhcm5pbmcgTGFZiZWwgUGFjayIsInF0eSI6MX0sljl4MTQiOnsidG10bGUiOiJlqKk5FVCoqIEVtbG1

0ZSBCaS1kaXJIY3Rpb25hbCBNZXRlcBFQ0EyLm4qliwicXR5ljoxfSwiMZA0MCI6eyJ0aXRsZSI6lkdpdi5BQyAzLjAgaW52ZXJ0ZXIiLCJxdHkiOjF9LCIzMDQ2ljp7InRpdGxlljoiR2I2RW5lcmd5IFdpRmkgRG9uZ2xlliwicXR5ljoxfSwiME1Nyl6eyJ0aXRsZSI6IkZhc3RlbnNvbCBibGFjayB1bml2ZXJzYWwgY2xhbXAiLCJxdHkiOjE4fSwiMzU4NCI6eyJ0aXRsZSI6Ikdyb3dhhdHQgU2hpbmVXaUZpLVgiLCJxdHkiOjF9LCIzNjU1ljp7InRpdGxlljoiR2I2RW5lcmd5IERDIE1pbmlhdHVyZSBDaXJjdWl0IEJyZWFrZXIgcKE1DQikiLCJxdHkiOjJ9LCI0Mjg0ljp7InRpdGxlljoiR2I2RW5lcmd5IDUuMmtXaCBMaUZIUE80IEJhdHRlcnkiLCJxdHkiOjF9LCI0NDg0ljp7InRpdGxlljoiR2I2RW5lcmd5IFBsdWcgdG8gTHVnIENhYmxlIChHZW4zKSIsInF0eSI6MX0sljQ0OTMiOmsidGI0bGUiOiJBQyBpc29sYXRvciAtIEINTyAtIDlwQSA0LXBvbGUiLCJxdHkiOjJ9LCI0NTU1ljp7InRpdGxlljoiQUmGaXNvbGF0b3lgLSBJTU8gLSA2M0EgNC1wb2xlliwicXR5ljoYfSwiNTEwNSI6eyJ0aXRsZSI6IlJ1YmJlciBHCm9tbWV0IGZvciBHZW4zIEludmVydGVyIFBsdWcgQ2FibGVzliwicXR5ljoxfSwiNTI2NSI6eyJ0aXRsZSI6IkppbmtvIFRpZ2VyIE5lbyA0NDBXIe4tVHlwZSBBbGwgQmxhY2sgTW9ubyBzb2xhciBwYW5lbnN0eSI6Nn0sljUyNzYiOmsidGI0bGUiOiJGYXN0ZW5zb2wgc2lsdmVyiHJhaWwgMzU1MG1tliwicXR5ljo3fSwiNTM5Nyl6eyJ0aXRsZSI6Ikdyb3dhhdHQgTUIOIDeWMDAwIFRMLVgyIFRocmVlIE1QUFQgMXBoIGludmVydGVyliwicXR5ljoxfSwiNTU2MSI6eyJ0aXRsZSI6lkdpdkVuZXJneSBHRU0xMjBDVCBnb2RidXMgRW5lcmd5IE1ldGVyliwicXR5ljoxfSwiljp7InRpdGxlljoiTWf0dCBTY2FmZm9sZGluZyBvdGVyIGNvbNlcnZhdG9yeSIsInF0eSI6MX0sljI2NDgjMzU5MCI6eyJ0aXRsZSI6IkZhc3RlbnNvbCBibGFjayBlbmQgY2FwliwicXR5ljoXmn19LCJ0cmFjZSI6eyJzcmMiOiJlYXN5LXB2liwibW9kZSI6InVrliwicHJvamVjdEIEljoiotg2MjA2In19",

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 #13

Created 8 September 2024 22:06:26 by Andy

Updated 16 July 2025 08:38:18 by Matt Agnes