

# Easy-PV API v0.2

## PV System Quote Tool

### Endpoints

For quote data generation : <https://soapi.easy-pv.co.uk/sandbox/design>

For quote report PDF : <https://soapi.easy-pv.co.uk/sandbox/report>

### Authentication

This API uses API keys to authenticate requests. Provide your API key within the 'x-api-key' header of your request. Without this your request will fail.

### Inputs - /design

You must send **POST** requests with an input object passed as JSON within the body of the request. The input object must contain the following parameters.

- *postcode*: Postcode of the building.
- *latitude*: Latitude of the building. Expected to be a number between 49 and 61.
- *longitude*: Longitude of the building. Expected to be a number between -11 and 2.
- *roofs*: An array containing object(s), each representing a roof. Each must have the following required properties:
  - *vertices*: An array of coordinates of the roofs vertices. The first two arrays define the roofs gutter.
  - *obstructions*: An array of obstructions, if present. Each obstruction is expected in the format ['type',width,height,x1,y1].
  - *pitch*: Angle of the roof, in degrees from horizontal. Expected to be a number between 0 (flat) and 90.
  - *roofType*: Type of roof. Expected to be one of the following values: "slate", "compositeSlate", "concreteTile", "plainTile", "panTile", "sheetMetal" or "flat".
  - *shading*: Percentage shading on the pv array as a decimal between 0 and 1.
- *annualConsumption*: The building's annual electricity consumption in kWh.
- *electricityImportTariff*: Electricity import tariff.
- *electricityExportTariff*: Electricity export tariff.

### Outputs - /design

The API will respond with a json encoded object with the following properties.

- *status*: The status of the request, which will equal 'success' or 'error'.
- *apiVersion*: A string representing the API version.
- *quotes*: An array of quotes, each containing the following properties:
  - *easyPVProject*: All the data required for Easy-PV to regenerate the quotes project, encoded into a string.
  - *battery*: A boolean which indicates the inclusion of a battery in the PV system.
  - *annualGeneration*: Annual PV generation in kWh.
  - *systemCost*: Total cost of the system in £.
  - *breakEvenYears*: Years until the break even date, or null if the number of years exceeds the projection period
  - *selfConsumption*: Percentage self consumption MCS estimate.
- *roofs*: An array of roofs, each containing the following properties:
  - *numberPanels*: The number of panels on the roof.
  - *panelModel*: A string containing the panel model.
  - *panelLayoutImage*: An SVG image of the PV system on the roof, stored as a string.
  - *panelLayout*: An object containing panel location data. Required by package roofmap.js to draw panels onto a canvas.

- *error*: Either false or an object containing the following properties:
  - *fatal*: A boolean representing the fatality of the error. A fatal error means no quotes could be generated.
  - *errorMessages*: An array containing strings which describe the error(s) in more detail.

## Example - /design

### JSON encoded parameters:

```
let input = {
  postcode: 'CB24 6AZ',
  latitude: 51.12345234,
  longitude: -0.022334,
  roofs: [
    {
      vertices:[[-5000,-2500],[-5000,2500],[5000,2500],[5000,-2500]],
      obstructions:[["velux",935,1342,1484,-1471]],
      pitch: 25,
      roofType: "slate",
      shading: 0.20,
    },
    {
      vertices:[[-5000,-2500],[-5000,2500],[5000,2500]],
      obstructions:[["chimney",935,1342,1484,-1471]],
      pitch: 35,
      roofType: "slate",
      shading: 0,
    }
  ],
  annualConsumption: 3100,
  electricityImportTariff: 0.146,
  electricityExportTariff: 0.051
};
```

### Usage with jQuery:

```
let settings = {
  "url": "https://soapi.easy-pv.co.uk/sandbox/design",
  "method": "POST",
  "headers": {
    "Content-Type": "application/json",
    "X-API-KEY": "xxxxxxxxxxxxxxxx",
  },
  "data": JSON.stringify(input),
};

$.ajax(settings).done(response => console.log(response));
```

### Response example:

```
{
  status: 'success',
  apiVersion: '0.2',
  quotes: [
    {
      easyPVProject: 'encoded string',
      battery: false,
      annualGeneration: 5199,
      systemCost: 1876.15,
      breakEvenYears: 6,
      selfConsumption: 0.50,
    },
    {
      easyPVProject: 'encoded string',
      battery: true,
      annualGeneration: 5199,
      systemCost: 2659.24,
      breakEvenYears: 11,
      selfConsumption: 0.60,
    }
  ]
}
```

```

],
roofs: [
  {
    numberPanels: 10,
    panelModel: 'Longi 315W all black split cell mono',
    panelLayoutImage: "SVG string",
    panelLayout: {panels:{}, blockInfos:[]}
  },
  {
    numberPanels: 5
    panelModel: 'Longi 315W all black split cell mono',
    panelLayoutImage: "SVG string",
    panelLayout: {panels:{}, blockInfos:[]}
  }
]
}

```

## Errors - /design

This API uses conventional HTTP response codes to indicate the success or failure of a request. Where possible, the API will return further error details in the body of the response.

An example of this is found below.

### HTTP Request error

```

{
  "status": "error",
  "statusCode": 400,
  "details": "Invalid UK Postcode."
}

```

A successful request can also return errors within the response object when a full quote could not be generated. A fatal error means no quote could be generated at all.

Examples of these can be found below.

### Fatal error

```

"error": {
  "fatal": true,
  "errorMessages": [ "Cannot fit any solar panels on the given roof(s)"]
}

```

### Non-fatal errors

```

"error": {
  "fatal": false,
  "errorMessages": ["Cannot calculate self consumption as system is not suitable for MCS calculations"]
}

```

```

"error": {
  "fatal": false,
  "errorMessages": ["No compatible batteries could be added to the system"]
}

```

## Inputs - /report

You must send **POST** requests with an input object passed as JSON within the body of the request. The input object must contain the following parameter.

- *easyPVProject*: All the data required for Easy-PV to regenerate the quotes project, encoded into a string. Same as that returned from the /design endpoint.

## Outputs - /report

The API will respond with a PDF file containing the quote report.

## Version History

This documentation is relevant for API version 0.2.

- **Version 0.1:** Released 18/09/2020. Documentation updated 24/09/2020 to include missing key-value pairs in the 'Inputs' section.
- **Version 0.2:** Released 13/11/2020. Some /design endpoint modifications, /report endpoint added.