FlexyGrid AI
Minimum Viable Product [MVP]
PV Production Forecast API

02/11/2023

Lead: TechBricks

Version: 0.1

I-NERGY project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 101016508

www.i-nergy.eu

Document History

Version	Date	Author [Partner]	Remarks
[1.0]	02/11/2023	Vladimir Cepric [TechBricks]	

Table of Contents

1. Introduction

This guide will provide you with all the necessary information on how to interact with the PV Forecast API, a RESTful service built using Flask, designed to provide solar energy production estimates.

2. Getting Started

The PV Forecast API is hosted on the https://ai.flexygrid.com/forecast/pv and listens on port 443.

3. Making a Forecast Request

Endpoint: /forecast_pv [Use the POST method]
Request Payload Structure:

```
{
  "latitude": "VALUE",
  "longitude": "VALUE",
  "declination": "OPTIONAL_VALUE",
  "azimuth": "OPTIONAL_VALUE",
  "kwp": "OPTIONAL_VALUE"
}
```

- **latitude [required]:** Latitude of the desired forecast location.
- **longitude [required]:** Longitude of the desired forecast location.
- **declination:** Angle of the solar panel relative to flat ground [optional].
- **azimuth:** Compass direction the solar panel faces [optional].
- **Kwp [required]:** Peak power of the solar system in kilowatts

Example Request Payload:


```
{
  "latitude": "54.9",
  "longitude": "25.3",
  "declination": "37 ",
  "azimuth": "0 ",
  "kwp": "1 "
}
```


4. Response Structure:

The API provides a detailed breakdown of solar energy production estimates.
Response Example:


```
{
  "result": {
    "watt_hours": {...},
    "watts": {...}
  }
}
```


- **result**: Contains the solar energy production forecasts, with breakdowns for hourly, daily, and overall periods.

5. Error Handling:

In case of any issues, the API has a robust error handling system.

Error Codes:

- **400**: Bad Request - The request was invalid or cannot be served.
- **404**: Not Found - The requested resource was not found.
- **500**: Internal Server Error - An unexpected error occurred on the server.

6. Computational Considerations:

The API's computational efficiency primarily hinges on the forecasting function and the evaluation against user-defined constraints. Although built on Flask and being I/O-bound, computational efficiency remains secondary to network latency, thanks to asynchronous requests and efficient operations via NumPy.

7. Resources:

Postman collection for test:
https://www.postman.com/devflexygrid/workspace/flexygrid-ai-i-nergy

Conclusion:

With the PV Forecast API, you can effortlessly retrieve insightful solar energy production forecasts, making it invaluable for energy management, grid planning, and resource allocation decisions. Ensure you provide accurate latitude and longitude values to get the best forecast results and remember that you can always tailor the analysis using user-defined constraints.