FlexyGrid AI
Minimum Viable Product [MVP]
Load 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** 3
2. **Getting Started** 3
3. **Making a Forecast Request** 3
4. **Response Structure:** 4
5. **Error Handling:** 5
6. **Computational Considerations:** 5
7. **Resources:** 5

Conclusion 5

1. Introduction

This guide will walk you through how to interact with our API and obtain accurate consumption forecasts based on your data.

2. Getting Started

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

3. Making a Forecast Request

Endpoint: /forecast_consumption [Use the POST method]

Request Format: Your request should be structured as a JSON payload containing the following fields:

- **data** required: Your time-series data for forecasting.
- **model**: Model choice. Defaults to '1' ARIMA.
- **n_periods**: Number of forecast data points. Defaults to 10.
- **freq**: Frequency of time series data, e.g., 'D' for daily.

Example Request Payload:

```
{
  "data": [["2023-10-01", 150], ["2023-10-02", 160], ["2023-10-03", 170]],
  "model": "1",
  "n_periods": 20,
  "freq": "D"
}
```

4. Response Structure:

Upon successful execution, the API will return a response with a breakdown of the consumption forecasts. This response will include dates and their corresponding forecasted values.

Response Example:

```
{
 "result": {
   "2023-10-04": 180,
   "2023-10-05": 190,
   "2023-10-06": 200,
   // ... continued for specified n_periods
 }
}
```

- **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.