

Backend Developer Application Assignment: Building a Data Processing API

Objective

Develop a backend application in Python that processes data from a provided JSON file representing assets, energy demands, energy outputs, energy systems, and their relationships. The application should expose an API endpoint to retrieve processed data based on specific queries.

Requirements

Data Access Object (DAO)

- Implement a DAO layer to load data from the provided JSON file.
- Organize and structure the data for efficient processing.

Processor

- Develop a processor layer to calculate energy-related metrics.
- The reduced energy demand is calculated as follows:
- For each energy type and energy system of an asset:
- If the energy system is electricity, subtract the energy output from the energy demand.

Controller

- Implement the controller layer using either Flask or FastAPI.
- Define a GET API endpoint requiring an asset ID in the request.
- The API endpoint should return the energy demand for the requested asset in the following format:

```
{
  "name": "",
  "energy_types": {
    "heat": 0,
    "water": 0,
    "light": 0,
    "cool": 0
  },
  "total_energy_demand": 0,
  "energy_output_reduction": 0
}
```

- The "name" field should contain the name of the requested asset.
- The "energy_types" field should include the energy demand for each energy type.

- The "total_energy_demand" field should represent the sum of all energy demands.
- The "energy_output_reduction" field should be a percentage, that indicates the difference between the initial energy_demand and the reduced energy_demand.
- Ensure the energy demand reduction is distributed equally and that energy_demand never goes below 0.

Documentation

- Provide clear documentation on how to run the application locally.
- Include instructions on making API requests and interpreting the returned data.

Coding Conventions/Guidelines

- Utilize descriptive variables for clarity.
- Comment your code for better understanding.
- Type variables and functions for improved readability.

Additional Guidelines

- Adhere to the provided coding conventions and guidelines.
- Organize the project into separate modules for DAO, processor, controller, and other necessary components.
- Utilize the provided JSON file efficiently for data processing.
- Prioritize efficiency and scalability when designing data processing algorithms.

Submission

- Submit your code repository (e.g., GitHub) containing the source code and documentation.

Evaluation Criteria

- Adherence to provided requirements and guidelines.
- Code quality, including readability, maintainability, and efficiency.
- Proper error handling and testing practices.
- Clarity and effectiveness of API design and documentation.
- Creativity and effectiveness in addressing encountered challenges.

Note: You may choose to implement the controller layer using either Flask or FastAPI. Please specify your choice in your submission.