FlatPeak

Backend Engineer Go Task

Design and code a simple REST API in Go that calculates time slots when carbon intensity is lowest.

Note: We expect the task to take around 4 hours. Please don't spend longer on it than that.

Carbon intensity is a measure of how much CO2eq is emitted per kWh and varies depending on how much renewable energy is available to the electricity grid. This is a simpler version of our Slots API which returns time slots when energy cost is lowest.

Data Source

There is a public forecast API for UK carbon intensity. Data is returned in 30 minute segments. You should calculate the slots based on the lowest forecast values.
https://carbon-intensity.github.io/api-definitions/#get-intensity-from-fw24h

Note: We use this API because forecast data is often a paid service. Access to energy data is often challenging!

Inputs

- Duration - Amount of time requested in minutes. Integer: defaults to 30 mins and errors if > 1440 mins (24 hours).
- Continuous - Boolean: when true, returns a single time period without breaks. When false multiple results may be returned as data is provided in 30 min time periods (defaults to false).

Requirements

- Accept and validate inputs.
- Fetch the forecast for the next 24 hours from the current time.
- Calculate the time slots when carbon intensity is lowest.
- Each slot result should include the forecast carbon intensity.
- If the slot spans multiple 30 minute periods you should return the weighted average of the forecast values.
- Partial periods like 45 mins should be supported and also return the weighted average.
- Output the results as JSON (see example API call below).
- Return an error response as JSON if calling the carbon intensity API fails.

Example API Call:

```
curl --request GET \
  -H 'Accept: application/json' \
  --url http://localhost:3000/slots?duration=30&continuous=true
[
  {
    "valid_from": "2025-01-09T01:00:00Z",
    "valid_to": "2025-01-09T01:30:00Z",
    "carbon": {
      "intensity": 142
    }
  }
]
```

Deployment

- Add a Dockerfile to build a docker image for your Go code.
- Your API endpoint only needs to be available locally.
- Add a README.md with how to run your solution and access the API using curl.

Sharing Code

- Email a tarball or zip file of your git repo to ross.fairbanks@flatpeak.com
- **OR** Publish your code on GitHub or BitBucket and send a link