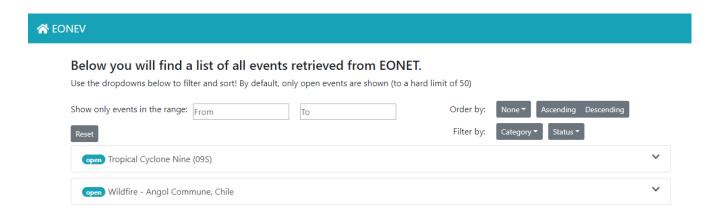EONEV

Earth Observatory Natural Events Viewer

Coding Challenge implemented by Yuri Antin Wergrzn

Readme and instructions: eonevapi github

Features

View list of events

Filter/order events by status

Filter/Order by date

Filter/Order by category

Style according to status

View/Expand individual Event

Back End Requirements

In Scope:
Connects to data source – EONET
Exposes event query endpoint with params
Handles ordering/filtering
Out of Scope:
Authentication
Data Storage
Sessions
Caching

Non Functional:
Use C#
JSON
Some documentation

Front End Requirements

In Scope:
Connects to back end API
Allows choice of sorting/filtering
Allows clicking on events to expand
Out of Scope:
Authentication
Sessions
Map visualization
Historical charts of events

Non Functional:
Uses React/TS
Easy to Use
Looks Nice

Customer

Renders App →

Inspiration:
https://www.onthisday.com/events/date/2019
https://www.justinmind.com/blog/list-ui-design/

React Web App

Returns Filtered Data →

1. Event Query
Input: …?category=a&status=b&dateStart=c&dateEnd=d
Output:
JSON containing root element events where
Event[i] is each event, with id, status, title, other metadata
2. Event Details
Input: event/{event_id}
Output: JSON containing event metadata

Back End API

Returns Raw Data →

https://eonet.sci.gsfc.nasa.gov/docs/v2.1#eventsAPI

EONET

About page

Front End → App

Event Viewer → Event List Controls

Notifies of filter changes

Event Viewer → Event List → Events (many)

Back end → Controllers → Services → Unit Of Work → Repositories → APIContext

eonevapi.api
Controllers
Exposes endpoints
to be called

eonevapi.services
Services
Performs filtering
and sorting logic

eonevapi.data
Unit Of Work
Abstracts access to
data

eonevapi.data
Repositories
Defines data
specific logic

eonevapi.data
APIContext
Performs HTTP call
to the API