

Weather Tracking Webtask

Matthew Lawson

Overview

The weather tracking webtask is hosted on webtask.io which allows users to register a desired zipcode through a REST API. All of the registered zipcodes will have their temperatures updated once every 5 minutes from the openweathermap.org API and the results are stored in Mongo. Users can then visit a web page which shows the most recent results of all the registered zip codes and is updated every 2 minutes.

The real purpose of course is to exercise several of the features of the webtask platform and demonstrate common Node activities. Some of the features utilized include:

- Use of Express
- Creating a REST API
- Creating a web view
- Integration with a third-party API
- Integration with MongoDB
- Use of the cron scheduler

Benefits

The benefits of the approach used, as opposed to a web page that does the weather API call on each reload, is that the weather API is called the minimum number of times, only once every 5 minutes for each zipcode. Also the zipcode requests are de-duplicated during the external query. This would be useful in the case where there were a high number of users hitting the webpage view, but each call to the external API was expensive in some sense to use.

Typical Use Case

To use this tool, a user would take the following steps:

1. Use the REST API to register a name and a zip code. The name is a unique identifier. The zip code may be duplicated.
2. Optional: Use the REST API to change their zip code for the registered name
3. Open a web browser to the page and watch breathlessly as the temperatures and timestamps occasionally change.
4. De-register their entry through the REST API.

Webtask Endpoints

GET /

This the web view. Example: Send your browser to:

`https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather`

GET /clients

List all registered clients along with recent weather data (if it's been queried). Example:

`curl 'https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather/clients'`

GET /clients/<name>

List the given client along with its recent weather data (if it's been queried). Example:

`curl 'https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather/clients/joe'`

POST /

This is trigger used by the cron scheduler to initiate the update. However it can be triggered manually. Example:

`curl -X POST 'https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather'`

POST /clients

Create or update a registration. The fields 'name' and 'zip' are required. Example:

`curl -X POST -H 'Content-Type: application/json' --data '{"name":"joe","zip":"90210"}' 'https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather/clients'`

DELETE /clients/<name>

Delete a registration. Example:

`curl -X DELETE 'https://wt-83c4a2eb13514f7221a7e7eeeb556174-0.sandbox.auth0-extend.com/weather/clients/joe'`

Limitations

The webtask works as a proof of concept, however the following items need improvement:

- There is no authentication done on the client
- Error handling is limited
- REST return status codes may not be correct in all cases
- REST return payload needs to be more formalized
- PUT is not implemented in favor of POST as a combo create/update
- Web page is bare bones