

IOOS Code Sprint Report Outs



Code Sprint Survey: <https://tinyurl.com/ioos-eval>
Please fill it out today!

Carpool Signup: <https://tinyurl.com/ioos-carpool>

ERDDAP Development and Configuration

[Link to Google Doc summary/notes](#)

Walkthrough of the ERDDAP code

Bob to add list of development priorities to ERDDAP Github repo

New development

- Allow user-derived variables in data configuration
 - This will solve the dateline crossover issues
 - Will also allow concatenation of columns in tabular data to automatically configure valid timestamps
- Demo the use of .dataTable format to easily create interactive graphics in the browser using Google charts
- Explore using existing ERDDAP formats to drive Vega visualizations (rather than integrating specific Vega visualizations into ERDDAP)
 - Other options are D3, TerriaJS, etc, etc, so stay flexible

ERDDAP Development

- Use of Google Analytics in ERDDAP
 - Micah mentioned RPS done something to use Google Analytics
 - Google Analytics won't catch the m2m usage through ERDDAP
 - Better to use tomcat logs with existing parsing packages
- Using ERDDAP 2.0 data ingest capability
 - HTTPS/SSL required
 - [Let's Encrypt](#) can help with testing
- Handling multiple languages
 - Point to the same data files but have two xml configurations in different languages.
- ERDDAPY
 - <https://github.com/pandas-dev/pandas/pull/28874> (allows us to pass any ERDDAP URL directly to pandas, including password protected servers via a session object.)

ERDDAP Configuration

- BCO-DMO ERDDAP config tool - Matt Biddle (coming soon-ish)
- Axiom/Kyle ERDDAP config tool - Jessica
 - <https://github.com/jessicaaustin/kyles-erddap-config-tool>
- Axiom ERDDAP fork
 - <https://github.com/axiom-data-science/erddap>
- Rich's approach for creating datasets.xml for a bunch of similar time series data (Python/Jinja2)
 - https://github.com/rsignell-usgs/erddap_tools
- NCCSV format in ERDDAP
 - [NetCDF-Compatible ASCII CSV File Specification](#)

To Do:

Bob will modify GenerateDatasetsXml to accept JSON object

IOOS "Gold Standard" dataset configurations will live here:

- <https://github.com/ioos/erddap-gold-standard>

ERDDAP installers should enable CORS by default, if possible

Cloud Migration Challenges and Solutions

Cloud Migration Challenges and Solutions

Participants:

Bob, Ben, Dalton, Alex, Ray, Emilio, Felipe

Morning Session (1 hr)

- **Discussion on Cloud Challenges:**

- Storage - costs and *fast* access (s3fs?)
- Cloud provider agnostic

- **Discussion on Cloud Solutions/Features:**

- Serverless (AWS Lambda)
- Container Orchestration (Kubernetes, ECS, ECR)

Afternoon Session (3hrs)

- Tutorial on AWS lambda features
 - Led by Dalton Kell
 - Lambda layers
 - Size limits, run time limits
- UDUnits validation API
 - Create lambda function to validate whether an input string is UDUnits compatible
 - Proved challenging to get UDUnits library into a lambda layer
 - Python package was fine
 - Can we use conda?

Cloud Migration Challenges and Solutions

Sprint:

AWS Lambda function to plot GliderDAC time-series profiles

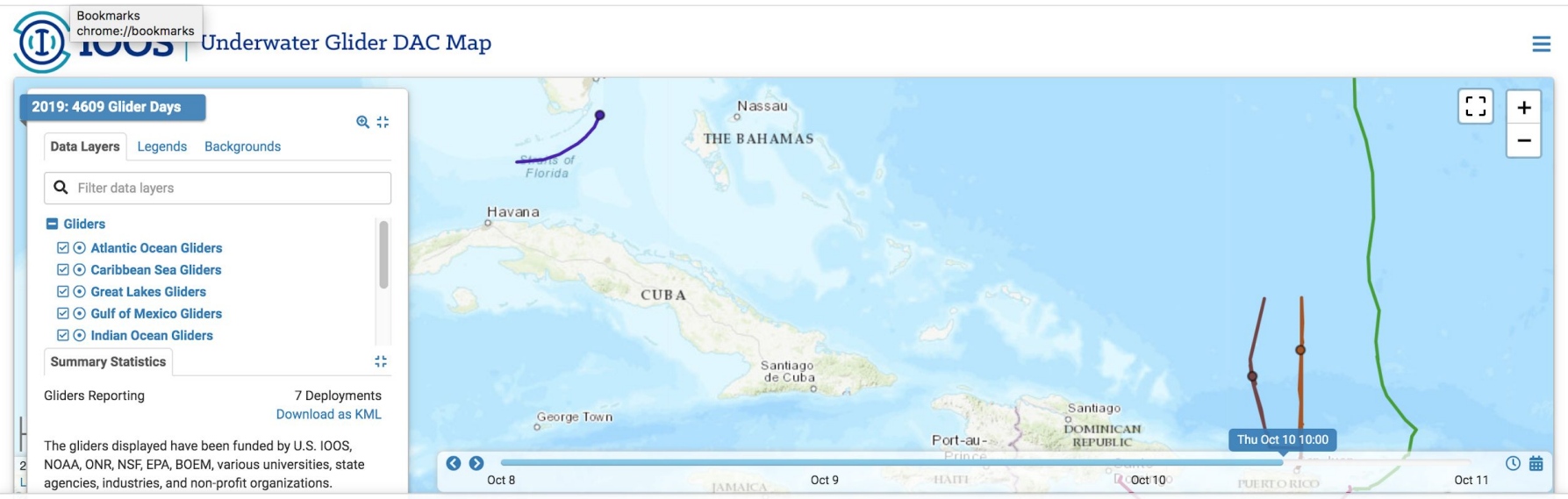
Use Case/Problem:

- GliderDAC time-series profile plots available from the GliderDAC map.
- Currently running as a cron job synchronously on all deployments every 6 hours.
- This is a poor implementation

Potential Solution:

- Lambda could make these plots asynchronous and event driven

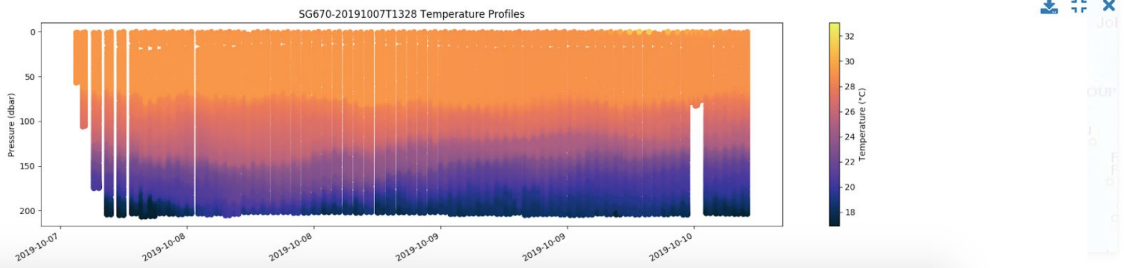
Cloud Migration Challenges and Solutions



Seaglider SG670 Deployed on 2019-10-07
Oct 7, 2019 09:33 (GMT -04:00) to Oct 10, 2019 06:57 (GMT -04:00)
 This project is funded by the National Oceanic and Atmospheric Administration (NOAA)

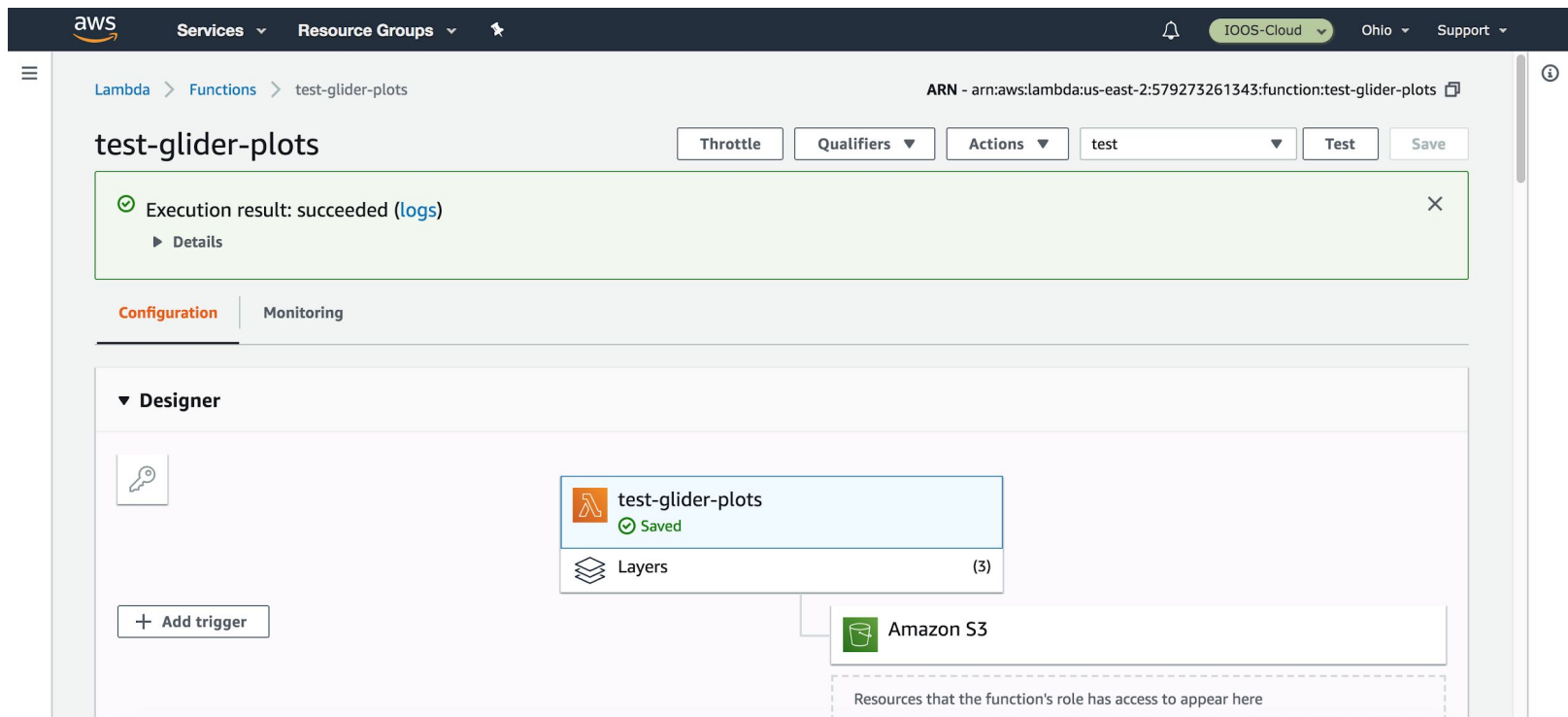
- Sea Water Density graph only
- Sea Water Electrical Conductivity graph only
- Sea Water Salinity graph only
- Sea Water Temperature** graph only

[View ERDDAP](#) [View THREDDS](#)



Cloud Migration Challenges and Solutions

- Lambda function: test-glider-plots
- Uses 3 lambda layers (netCDF4, matplotlib, AWS scipy)
- Give it access to an S3 bucket to upload plots



The screenshot displays the AWS Lambda console for the function 'test-glider-plots'. At the top, there is a navigation bar with 'aws', 'Services', 'Resource Groups', and a notification bell. The function's ARN is shown as 'arn:aws:lambda:us-east-2:579273261343:function:test-glider-plots'. Below the function name, there are controls for 'Throttle', 'Qualifiers', 'Actions', a dropdown menu set to 'test', and 'Test' and 'Save' buttons. A green notification box at the top left states 'Execution result: succeeded (logs)' with a 'Details' link. The 'Configuration' tab is active, showing the 'Designer' view. The function is marked as 'Saved' and has 3 layers. An Amazon S3 bucket is connected to the function. A dashed box at the bottom indicates 'Resources that the function's role has access to appear here'.

Cloud Migration Challenges and Solutions

- Invoke the lambda function via boto3 library in python
- Integrate into GliderDAC server processing by leveraging ERDDAP dataset pub/sub

The screenshot displays the AWS S3 console interface. The breadcrumb path is 'Amazon S3 > ioos-code-sprint-2019'. The 'Properties' tab is selected. A search bar contains the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. A table lists the contents of the bucket:

Name	Size	Last Modified	ETag
SG670-20191007T1328	--	--	--
ru01-20140120T1444	--	--	--
ru01-20140123T1250	--	--	--
ru01-20140217T1244	--	--	--
ru22-20130924T2010	--	--	--
ru23-20130305T2004	--	--	--
ru23-20130910T1701	--	--	--
ru23-20131017T1614	--	--	--
ru23-20140814T1239	--	--	--

An inset heatmap chart titled 'SG670-20191007T1328 Temperature Profiles' is overlaid on the right side of the console. The y-axis is labeled 'Pressure (dbar)' and ranges from 0 to 200. The x-axis shows dates from 2019-10-07 to 2019-10-10. A color scale on the right indicates temperature in degrees Celsius, ranging from 18 to 32. The chart shows a clear diurnal cycle in temperature, with warmer temperatures (orange/red) during the day and cooler temperatures (blue/purple) at night.

QC/QARTOD

QARTOD Library Implementation

Full list of tasks and notes in [the google doc](#)

- **Consolidated repo now live at**
http://github.com/ioos/ioos_qc (conda-forge/ioos_qc)
 - 8 other repos archived or deleted, old pkgs deprecated
 - Will release ioos_qc 1.0.0 shortly to conda-forge and pypi
- Repo reviewed by 6+ people
 - Meets Felipe's "Must Haves" and "Nice to Haves"
 - Bugfixes made and issues created
 - Beefed up the documentation and notebook examples
 - Re-do implementation of Attenuated Signal Test
 - Lots of work to improve NcQcConfig
 - Started list of non-python repos for reference
- Extensive discussion on best practices for implementing QARTOD (will go in ioos_qc docs)
- User Stories for QC Config Management Tool

Bio Data Management

Bio Data Management

Resources: [notes](#), [dir](#), [Github repo](#)

Learning

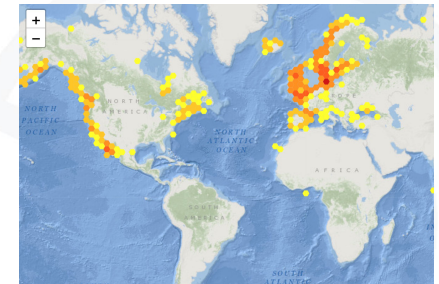
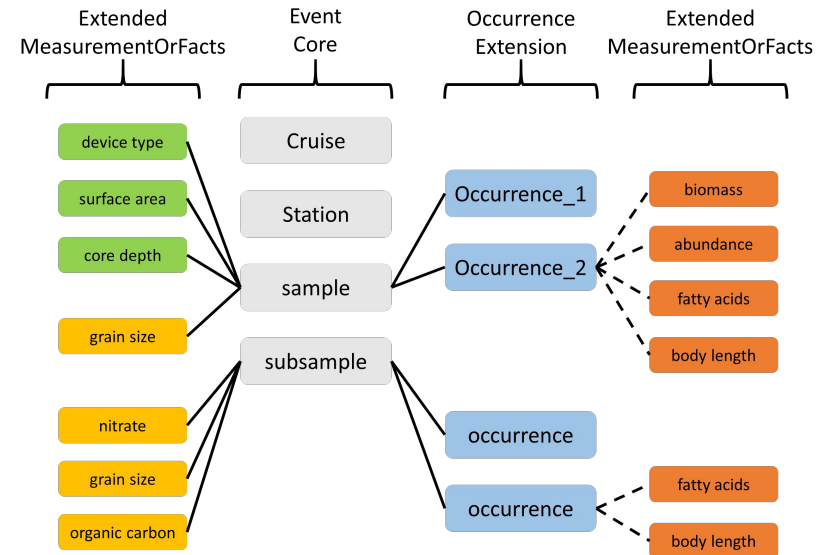
- What is Darwin Core and why should you care?
- What is OBIS and why should you use it?

Doing

- Transformed biological datasets to DwC

Using

- Map obs via OBIS/GBIF, spp dist mdl
- Archive, transform, discover:
DataOne, ERDDAP, OBIS (bioschemas.org?)



Data Cataloging/Data Discovery

Data Cataloging/Data Discovery

Purpose: Help hypothetical “civic hacker” **discover** datasets and **retrieve** data without learning a lot about CKAN/CSW/ERDDAP/TDS ...

Discovery: Query IOOS Catalog what’s near a given lat long, return list of datasets with lists of parameters.

Retrieval: Use IOOS catalog resource metadata to formulate “appropriate” ERDDAP or THREDDS URLs.

Next: Consider IOOS Catalog API capabilities and decide whether to develop or document.

Cool Stuff: [Super-rapid API prototyping](#) using Swagger/OpenAPI (thanks, Ben Adams).

IOOS Client Libraries/GitHub Management

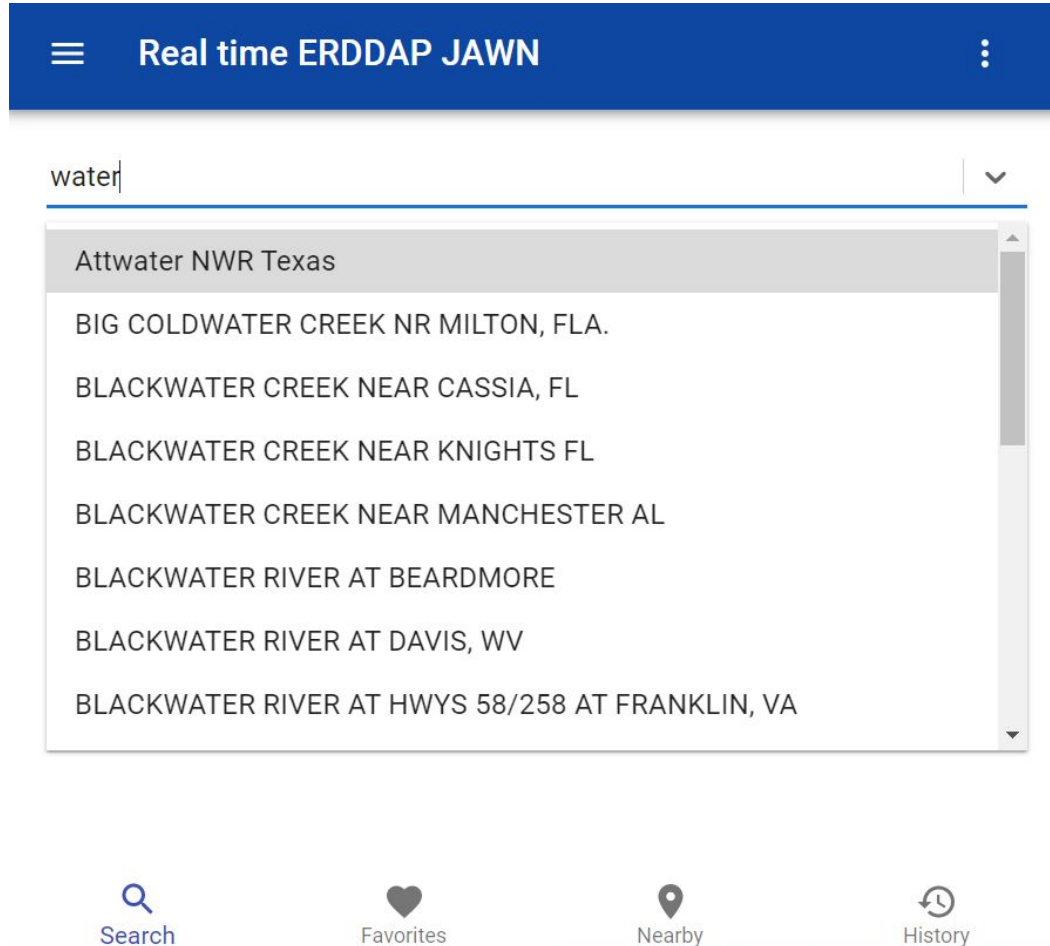
IOOS Client Libraries/GitHub Cleanup

- Partial GH cleanup complete!
 - <https://hackmd.io/rYrSEAybSKKHMsvHNidGMg?view>
- Path forward for pyoos

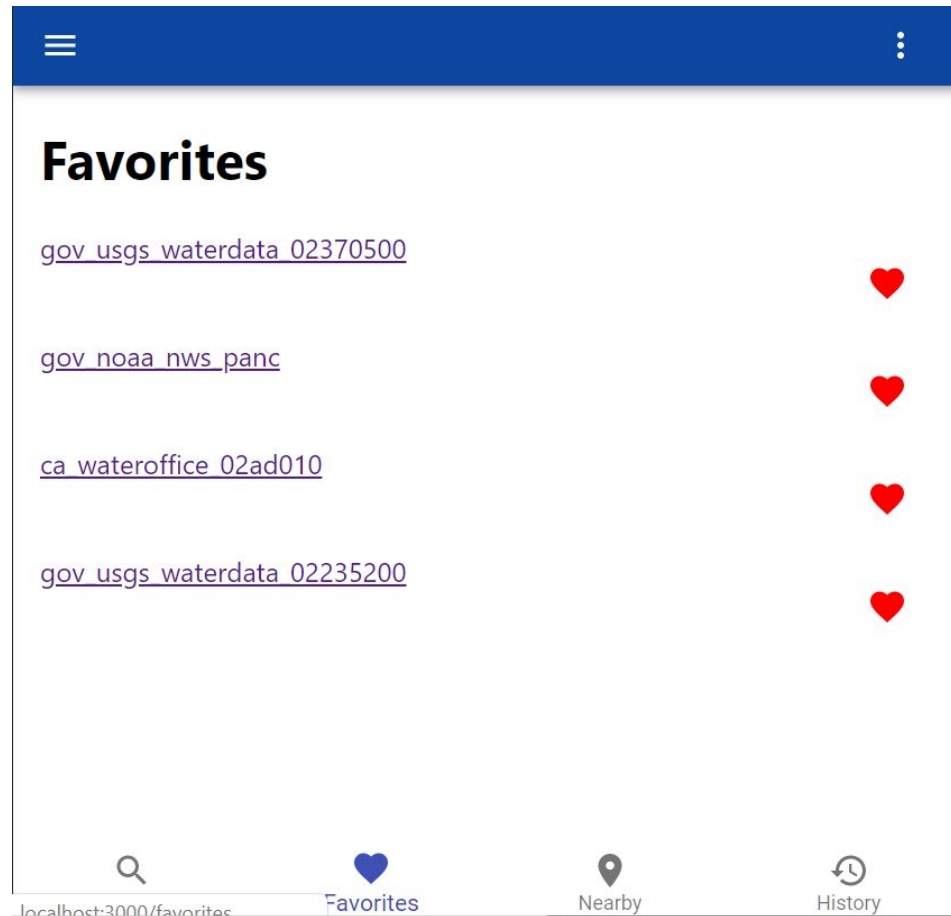
Common IOOS Mobile/Web App

- Tests
- [Storybook](#)
- Dave got his T-Shirt
- Demos
 - [App](#)
 - [Glider curtain plot](#)
- Repos
 - [erddap-timeseries-chart](#)
 - [erddap-parser](#)
 - [erddap-realtime-app](#)

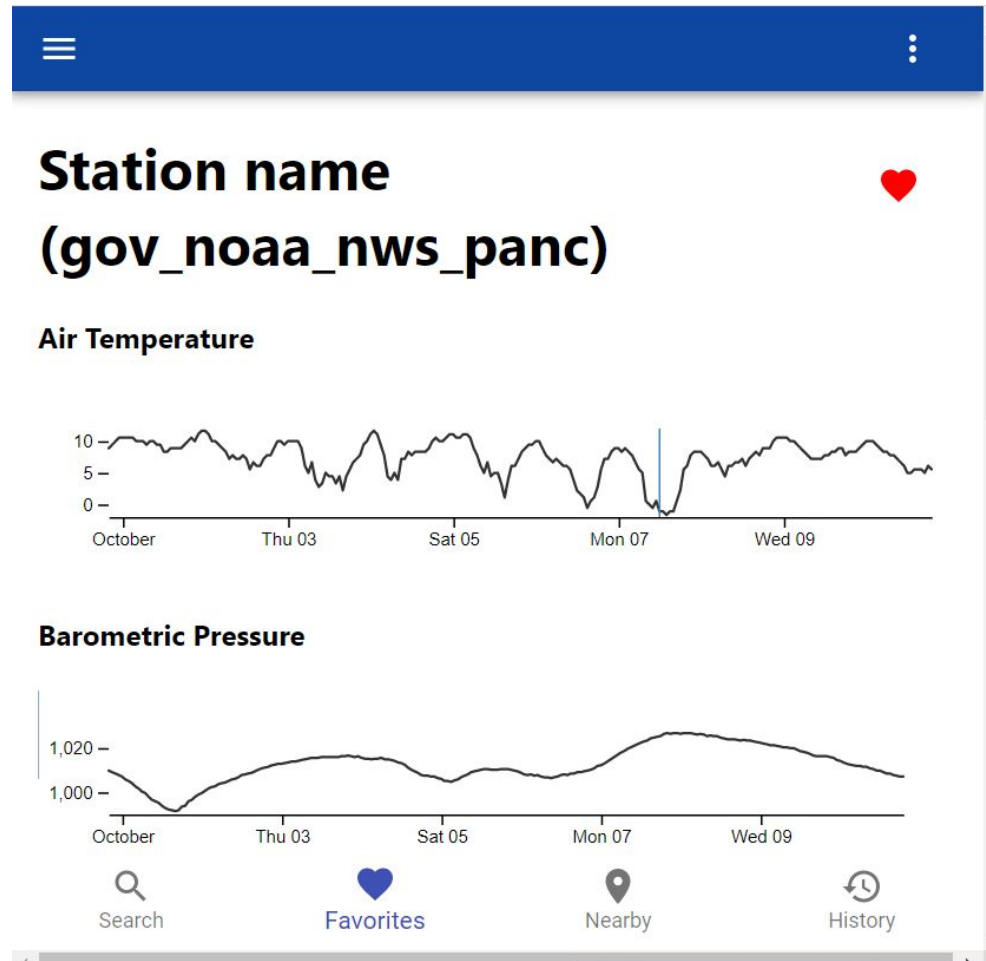
Search



Favorites



Station display



Curtain plots

