DWRAT Web Application Setup Guide

This document is meant to provide a description of the DWRAT web application and information sufficient for technical SWRCB personnel to migrate the application to SWRCB’s servers. Questions on migration of the server or application setup can be addressed to Nick Santos at [nrsantos@ucdavis.edu](mailto:nrsantos@ucdavis.edu).

Table of Contents

[Setting up the Server 1](#_Toc497314166)

[Requirements 1](#_Toc497314167)

[Dependencies 1](#_Toc497314168)

[Hosting 2](#_Toc497314169)

[Migration to SWRCB Servers 2](#_Toc497314170)

[Maintenance 2](#_Toc497314171)

[Loading New Data Into the Application 3](#_Toc497314172)

[The master spreadsheet 3](#_Toc497314173)

[Adding an item 4](#_Toc497314174)

[Continuing Application Development 4](#_Toc497314175)

[Development Commands 5](#_Toc497314176)

# Setting up the Server

## Requirements

This code base comes precompiled into a client-side Javascript web application. As such, it only requires a standard web server capable of hosting a directory of files in order to be served.

## Dependencies

The built version of this software has no software dependencies to run. However, in order to reliably serve data, it does rely on data hosted in three external locations:

1. Google Docs
2. Center for Watershed Sciences ArcGIS Server
   * <http://atlas.cws.ucdavis.edu/arcgis/rest/services/Watersheds/>
     + WMS and WFS should be enabled
     + Currently used in app.js, line 518 for var lookupUrl and app.js line 1338 in var wms2
   * <http://atlas.cws.ucdavis.edu/arcgis/rest/services/CalWater_Hydrologic_Unit/>
     + Standard ArcGIS services enabled
     + Currently used in app.js line 1325 for var wms1
3. SWRCB ArcGIS Server
   * <http://gispublic.waterboards.ca.gov/arcgis/rest/services/Water_Rights/Points_of_Diversion/>

Services hosted on CWS infrastructure can be passed to SWRCB for hosting on their ArcGIS servers. The aforementioned features must be enabled for each service, and the code should be edited at the mentioned locations to correctly reference the new services.

The load on these services is minimal, at least in our testing – the applications sends textual queries to the *Points\_of\_Diversion* service and displays the other two on the map – these services can be cached for speed to reduce load on the server when users aren’t querying the underlying data. The web application was publicly available at CWS, but unlisted on our website or the Internet, so it acted as a private application. If SWRCB uses it at the rate it has been used previously, server load is not a significant consideration. For broad public deployment or significant internal usage, load may need to be reassessed.

## Hosting

The built application lives in the dist directory of this repository. To host the application, clone either the [entire GitHub repository](https://github.com/ucd-cws/dwart-app/), or just the contents of the dist directory into a folder on your server. The only files you need to host are the contents of the dist directory - everything else is used for development and building of the application. The code is standalone once served by any standard web server – it does not require special application servers, databases, or server-side technologies. Everything runs in the web browser.

# Migration to SWRCB Servers

To sum up the steps to deploy this application on SWRCB servers, all that is required is to:

1. Copy the contents of the “dist” folder from the application’s repository to a directory that is served on the web
2. Copy the two ArcGIS Server services from CWS’s server to SWRCB server
3. Edit app.js to use the new SWRCB services:
   * app.js, line 518, edit var lookupUrl to have the URL to new Watersheds service
   * app.js, line 1338 edit var wms2 to have the URL to new Watersheds service
   * app.js, line 1325 edit var wms1 to have the URL to the new *CalWater\_Hydrologic\_Unit* service

## Maintenance

Since the application doesn’t require external services, ongoing maintenance is minimal. The application is developed with modern web technologies, in part to ensure the lifespan of the application is maximized. Web browser vendors occasionally do make changes that impact pages. For example, many legacy applications needed updating when browser vendors stopped loading nonsecure (http) content on secure pages (https). In that case, behind the scenes and code changes were both needed to bring applications back online.

For this application, the more likely issue requiring maintenance would be either changes to the hosted ArcGIS services or to Google’s API for accessing spreadsheets. None of these require ongoing maintenance, but staff should pay attention to changes in either of those places to know if updates are required. As an example, between completion of this application and delivering this setup guide, a change to SWRCB’s Point of Diversion ArcGIS Service broke the application. Coordination between responsible units may be necessary to keep the web application functional.

# Loading New Data Into the Application

This application has two primary sources of data. The user can upload files with new runs from DWRAT to visualize in the application or the user can select a permanently saved run of the application from the Public Directory. For temporary visualization for staff running DRWAT the upload option is best, but for other uses, including saving runs, visualization by other staff not involved in running the DWRAT model, or the general public, adding items to the Public Directory is the best way to make data available. When using the application, these items can be accessed by clicking Public Directory in the top right corner, then selecting the relevant filename to load for display.

It's important to note that the web application has no knowledge of SWRCB internal servers or services and does no synchronization with these services on its own. Any data loading, as currently coded, must be manually triggered, following the information below.

## The master spreadsheet

Datasets available in the Public Directory are indexed in [the "DWRAT Runs" stored in Google Sheets](https://docs.google.com/spreadsheets/d/1ACi7P0pOy-JRjtw6HribQphhEOZ-0-CkQ_mqnyCfJcI/). For administrators of the site, if the spreadsheet used for the Public Directory index needs to be changed, such as during transition from CWS to SWRCB, do the following:

1. Make a new Google Sheet
2. Add the following fields to this sheet in the first row:
   * *Region*: The name that shows up when selecting items from the Public Directory in the application
   * *Start Date*: M/DD/YYYY format date that each record's information starts on
   * *End Date*: M/DD/YYYY format date that each record's information ends on
   * *ID*: The ID of the Google Sheet that holds the data for this record, as discussed in *Adding an Item* above.
3. Share the sheet so that it is publicly viewable, but not publicly editable. It is fine if it is unlisted, but people with the URL should be able to load it.
4. Edit the sheet URL into the code by finding the line where public\_dir\_root is defined - it will start with var public\_dir\_root. As of this writing, the line is in app.js on line 1076, but that is subject to change in future builds.
5. Relaunch the application

To share with staff so they can add an item to the public directory, they can send details (information below) to the creator of the spreadsheet, the creator of the spreadsheet could share the sheet with them in Google Docs with edit permission, or staff can request access on their own. To request acces, staff can load the spreadsheet and click the icon that says "View Only" in the top right corner then click "Request Edit Access". The spreadsheet’s creator will receive an email with the request and once approved, the requesting staff will have the access needed to add items to the Public Directory.

## Adding an item

To add an item to the public directory, upload the data as a new sheet. Share the spreadsheet publicly (for viewing only), then copy the ID in the URL of that spreadsheet for use in the master spreadsheet. For example, if the URL of the uploaded spreadsheet is

https://docs.google.com/spreadsheets/d/1ACi7P0pOy-JRjtw6HribQphhEOZ-0-CkQ\_mqnyCfJcI/

then the ID you need is

1ACi7P0pOy-JRjtw6HribQphhEOZ-0-CkQ\_mqnyCfJcI

If the URL has any more information following that ID, exclude it - only use the portion mentioned here.

From there, create a new record in the master spreadsheet. Place that ID in the ID column of the record, then enter the other details you want to show in the public view of the spreadsheet, including a name, start date, and end date. Once you have added the record, the new data will be available in the Public Directory portion of the application.

# Continuing Application Development

Should SWRCB wish to further develop the application, the original source code used to build the dist directory is available on GitHub at <https://github.org/ucd-cws/dwrat-app>. The dist folder used for serving the application is a product of the rest of the code in the repository.

To edit the application, it is best to work with the other code in the repository. The code is written in Javascript using NodeJS/CommonJS Modules and we build the application by running it through Grunt and Browserify to create a browser friendly package. While not a complete tutorial, the following commands will be useful while developing.

To develop the code, you will need to install NodeJS, available at <https://nodejs.org> – after installation you can run the following commands

## Development Commands

npm install bower

Will install a required package manager used by the remaining commands

npm run init-dev

will install dependencies from both Node Package Manager (NPM) and Bower.

npm run dev

Runs **watch** process for testing code while developing. This watches for changes to the lib/shared js files and builds a new public/js/app.js file automagically after you make a change

npm start

Serves /public on the local computer so that changes to code can be tested and visualized in the browser. Not for use on a production server

npm build

Builds the /dist folder from the current code. This new dist folder can then be deployed as described earlier in this guide

npm run start-dist

Serves /dist for testing. Not for production use.