

Introduction

Now let's take what you've learned so far and apply it to our course project. For the course project, we'll be creating an interactive visualization showing on-time arrival and departures for US airports for the December 2014.

Project Goals

There are a couple of goals we want to achieve with this project.

- We want to show all the airports on the map at the same time.
- We want to be able to easily identify the best and worst airports based on on-time arrival and departures. We will accomplish this by using four different colored (green, yellow, orange and red) icons that indicate the airport performance from best to worst.
- We want to visualize the largest and smallest airports based on total operation or flight volume. In our project, we will group airports by total operations based on three different icon sizes, one and a half times the size, one time, and a half a time. So, larger icons will indicate the larger airports, while smaller ones will indicate the smaller airports.
- The final goal is to create a styled info window that when you click on an airport, you can view all the details. In the info window, we want to show the average on-time percentage, along with individual percentages for the arrivals and departures. We also want to include the airport code, city and state, and coordinate location.

Data for course projects

To get started, we first need the data source for our project. The data is available from the Department of Transportation website in a PDF report. You can visit www.dot.gov/airconsumer/february-2015-air-travel-consumer-report to look at the data.

If you scroll down to page 16, the report includes a list of US airports. Each airport has the airport code, city and state, on-time arrival and departure percentage, and total operations or flights for both arrival and departures. We need to take the data in this PDF and create a JavaScript object that we can parse to add the airports to the map.

To extract this information, I copied and pasted the data from the PDF into my Brackets text editor. I then used the column editing feature to create a JavaScript object and added each key to the value pair.

We also need latitude and longitude location for each airport in order to display it on the Google map. To get the location information, I geocoded the city, state and airport to get a latitude and longitude that I then appended to the data.

Display Info Window on marker click

I've stored the data as a JavaScript object that you can find in the project in the file *airportdata.js*. The object contains an array of airports, and each airport element contains eight different key-value pairs to store the information we pulled from the PDF, as well as the latitude and longitude location.

Let's look more closely at the information stored for each airport:

- The first key is the airport, and that stores the city, state, and airport code.
- Next there is the code, which represents the airport code.
- Then the location. `lat` and `lng` represent the latitude and longitude of that particular airport.
- Next we have the arrival and departures on-time percentage. Those are represented by `aper` and `dper`.
- Finally, we have the total arrival and departure flights. `aop` represents the arrivals, and `dop` represents the departures.

For this project, I chose to store the data on the client-side in a JavaScript object so we could focus on the Google Maps API. The format of the data is JSON, so it represents a typical data set that might be retrieved from a server using an AJAX call. So let's jump into the code and start creating our airport map. We will start with the code we finished in the last lesson.

