9.5.4

Stations Route

Remember all of the work you did on the stations analysis? Now you'll build a route for your app that will allow this analysis to come to life.

You completed two routes, so now it's time to move on to the third: the stations route. For this route we'll simply return a list of all the stations.

Begin by defining the route and route name. As a reminder, this code should occur outside of the previous route and have no indentation. Add this route to your code:

```
@app.route("/api/v1.0/stations")
```

With our route defined, we'll create a new function called `stations()`. Go ahead and add the following code:

```
def stations():
    return
```

Now we need to create a query that will allow us to get all of the stations in our database. Let's add that functionality to our code:

```
def stations():
    results = session.query(Station.station).all()
```

```
    return
```

We want to start by unraveling our results into a one-dimensional array. To do this, we want to use the `function np.ravel()`, with `results` as our parameter.

Next, we will convert our unraveled results into a list. To convert the results to a list, we will need to use the list function, which is `list()`, and then convert that array into a list. Then we'll jsonify the list and return it as JSON. Let's add that functionality to our code:

```
def stations():
    results = session.query(Station.station).all()
    stations = list(np.ravel(results))
    return jsonify(stations=stations)
```

NOTE

You may notice here that to return our list as JSON, we need to add stations=stations. This formats our list into JSON. If you'd like to read more about it, checkout the Flask documentation (https://flask.palletsprojects.com/en/2.0.x/api/#flask.json.jsonify) .

The stations route is ready to be tested! To test it, run the code in the command line and then check if the result is correct in the web browser (`http://localhost:5000/`). Don't forget to add the remainder of the route to see the output of your code. Here's what your results should look like in the web browser:


```
{"stations":
["USC00519397","USC00513117","USC00514830","USC00517948","USC005
18838","USC00519523","USC00519281","USC00511918","USC00516128"]}
```


If your output is not the same as above, make sure to double-check your code to ensure you didn't miss anything.

Once you've got everything looking correct, you are ready to move on to the temperature observations route.