# Coding Challenge: Weather

Below are the details needed to construct an ATM/branch locator application based on a public JPMC API.

**Public API**
Create a free account at openweathermap.org. Just takes a few minutes. Full documentation for the service below is on their site, be sure to take a few minutes to understand it.

http://api.openweathermap.org/data/2.5/weather?q=Westervile,oh,us&APPID=YOUR_APP_ID

You will also need icons from here:

http://openweathermap.org/weather-conditions

**Requirements**

These requirements are rather high-level and vague. If there are details I have omitted, it is because I will be happy with any of a wide variety of solutions. Don't worry about finding "the" solution.

1. Create a browser or native-app based application to server as a basic weather app.
2. Search Screen
    1. Allow customer to enter a US city
    2. Call the openweathermap.org API and display the information you think a user would be interested in seeing. Be sure to has the app download and display a weather icon.
3. Auto load the last city searched upon app launch.

In order to prevent you from running down rabbit holes that are less important to us, I'll try to prioritize what we are looking for versus what is less meaningful.

**What is Important**

- Proper function – requirements met.
- Well constructed, easy-to-follow, commented code (especially comment hacks or workarounds made in the interest of expediency (i.e. // given more time I would prefer to wrap this in a blah blah blah pattern blah blah )).
- Proper separation of concerns and best-practice coding patterns.
- Defensive code that graciously handles unexpected edge cases.

**What is Less Important**

- UI design – generally, design is handled by a dedicated team in our group.
- Demonstrating technologies or techniques you are not already familiar with (for example, if you aren't comfortable building a single-page app, please don't feel you need to learn how for this).
- Showing off in-depth knowledge of JavaScript semicolon syntax requirements.

**Bonus Points!**

- Automated tests!

- Good design (I know I said it was less important, but what I mean is I don't want a beautiful, poorly constructed app).
- Additional functionality – whatever you see fit.

As I mentioned, you are not expected to function in a vacuum. Use all the online resources you can find.