TEKLINKS®

We Make IT Work for Business.

Why Devops?

The new digital business / city / country

Digital business is the creation of new business designs that connect people, business and things to drive revenue and efficiency.

Gartner

Transform Processes & Business Models

Leads To:

Innovation

Faster Time to Market

Empower Workforce Efficiency and Innovation

Leads To:

Increased Productivity

Better Retention

Personalize Customer/ Citizen Experience

Leads To:

Increased Loyalty

Greater Insight

Bridge the Digital Gap in Lock-Step.

And Understand that the Gap is WIDE for Most!

Technology

21%

Of organizations feel they have the right technology in place to succeed in Digital

Teamwork

14%

Of organizations feel they have the right processes in place to succeed in Digital

Talent

16%

Of organizations feel they have the right talent in place to succeed in Digital

Source: State of Digital Business, 2015-2020, Forrester, 2016

API Intro

What is an API?

Application Programming Interface

Application program interface (API) is a set of routines, protocols, and tools for building software applications. An API specifies how software components should interact.

– Vangie Beal @ webopedia.com

APIs You Might Have Seen

- Windows DLLs
- JTAPI drivers
- WSDL/SOAP
- REST
- Others?

APIs Require A Common Data Format

- CSV
- EDI
- XML
- JSON

Focused on REST and JSON

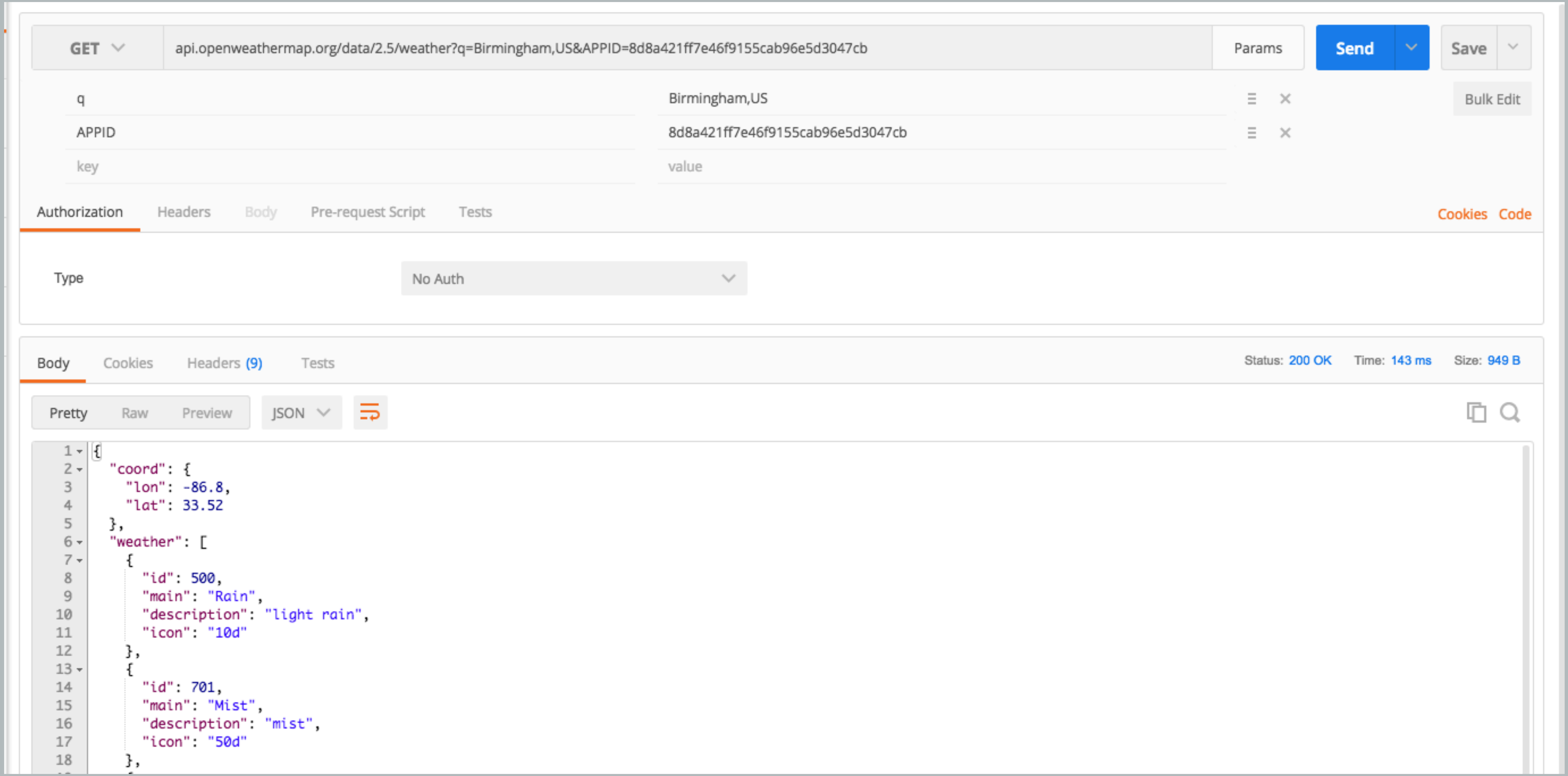
- Most modern Web APIs implement a REST interface
- Most modern Web APIs support JSON as a data format

What is REST?

- **RE**presentational **S**tate **T**ransfer
- **REST**ful Web Services
- Uses HTTP Verbs - GET, POST, PUT, PATCH, DELETE, etc...
- Accessed via URI's that point to a particular endpoint
  - Open Weather Map's REST Endpoint to get the Weather
  - http://api.openweathermap.org/data/2.5/weather
  - http://api.openweathermap.org/data/2.5/forecast

Parameters?

- For these endpoints to return meaningful data, you often need to ask for something specific
- Parameters can be passed as query parameters in the URI
  - http://api.openweathermap.org/data/2.5/weather?**q=Birmingham,US**
  - "q" is the parameter and "Birmingham,US" is the value we are looking for
  - Multipe parameters can be passed with the ampersand "&"
  - .../2.5/weather?q=Birmingham,US**&**APPID=8d8a421ff7e46f9155cab96e5d30....
- A good API will be well documented to make it easy to use
  - https://openweathermap.org/current



GET api.openweathermap.org/data/2.5/weather?q=Birmingham,US&APPID=8d8a421ff7e46f9155cab96e5d3047cb

q	Birmingham,US
APPID	8d8a421ff7e46f9155cab96e5d3047cb
key	value

Type: No Auth

Status: 200 OK Time: 143 ms Size: 949 B

```
{
  "coord": {
    "lon": -86.8,
    "lat": 33.52
  },
  "weather": [
    {
      "id": 500,
      "main": "Rain",
      "description": "light rain",
      "icon": "10d"
    },
    {
      "id": 701,
      "main": "Mist",
      "description": "mist",
      "icon": "50d"
    },
```

JSON - JavaScript Object Notation

- Standardized in 1999
- Used to exchange data between APIs
- Easy to read
- Easy for machines to parse
- Can store objects, arrays, numbers, strings, booleans, and null

JSON Formats

- Two basic formats:
    - Single objects
    - Arrays

Single Object

- A single object with an unordered set of name/value pairs. It begins with a left brace "{" and ends with a right brace "}"
- All strings must be double-quoted

```
{
        "firstName": "Jeremy",
        "lastName": "Sanders",
        "email": "jsanders@teklinks.com",
        "children":[ "Sam", "Olivia", "Everett"],
        "age":39,
        "wife":{
            "name": "Kandy"
        }
}
```

Array

- An ordered list of objects. It begins with a left bracket "[" and ends with a right bracket "]"
- All strings must be double-quoted

```
[
        {
            "id":1,
            "name": "Jason",
        },
        {
            "id":2,
            "name": "Jeremy"
        }
]
```

Open Weather API Response

```
{
  "coord": {
    "lon": -86.8,
    "lat": 33.52
  },
  "weather": [
    {
      "id": 500,
      "main": "Rain",
      "description": "light rain",
      "icon": "10d"
    }
  ],
  "id": 4049979,
  "name": "Birmingham",
  "cod": 200
}
```

Platforms/Languages/Tools

- Ok, so how do you do this?
- Microsoft .Net (VB, C#)
- Python
- Ruby
- Java
- **JavaScript** and **NodeJS**

JavaScript

- I thought that was for client side browser stuff.
- It's for the server too.
- We need to know some basics:
  - Datatypes
  - Operators
  - Control Structures
  - Objects
  - Functions
  - End each line with a semi-colon

Datatypes

- String - single or double quote enclosed string of Unicode characters (ie: "Jeremy")
- Number - integers, doubles, singles, it's all the same
- Boolean - true/false
- Objects - This can be an object based on a JavaScript class
- Functions - a custom function is assigned a variable name too
- Array - this is an ordered list of other variables
- Date - they get their own type
- Null
- Undefined (different than null)

Operators

- Numeric operators +, -, *, /, and % (remainder)
- Assignment operator =, +=, -=
- Additive operator ++, --
- Concatenation operator + (didn't we already use that one)
- Comparison operators <, >, <=, >=, ==, ===, !=, and !===

Control Structures

- if/else

```
if (hal=='tall') { sky.normal } else { pigs.fly };
```

- while loops

```
while (jason=='skinny') { jason.eat };
```

- for loops

```
for (var i=0; i < 5; i++) { mark.offensiveComment };
```

- switch

```
switch (client) { case 'casino': sendto(joel); }
```

Objects

- An object is basically an associative list of name/value pairs (Remember JSON Objects)

```
var engineer = {
    name: "Powell",
    favSoftware: "ISE",
    favSport: "football",
};
```

Functions

- A function can take named parameters as input, perform actions, and return a value at its termination

```
//use a function to prototype an object
function newEngineer(name, favSoftware, favSport) {
    engineer.name = name;
    engineer.favSofware = favSoftware;
    engineer.favSport = favSport;
    return engineer;
}

//call the function to create a new instances of that object
var pdavis = newEngineer("Powell", "ISE", "football");

console.log (pdavis.name); //what will this print?
```

End Each Line with a Semi-Colon

JavaScript Tools We Will Use

- NodeJS is a JavaScript server
  - Has thousands of opensource modules available for use
  - Is scalable and yet easy to use
- Text editor of your choice (VSCode, Notepad++)
  - Syntax highlighting
  - Code Auto-completion
- Postman - client for HTTP requests
  - Helps you fine tune your REST requests to external APIs
  - Allows you to test your own API

Tools Part 2

- ngrok - Allows you to "securely" expose a webserver running on your local machine to the internet
- Github/BitBucket - code repositories

Other Fundamentals

- HTTP Response Codes
  - http://www.restapitutorial.com/httpstatuscodes.html
- HTTP Verbs
  - http://www.restapitutorial.com/lessons/httpmethods.html
- HTTP Authentication Types
  - Basic - Username and Password with a Base64 hash

    ```
    var req.auth = "Basic " + base64("Jeremy" + ":" + "password123");
    ```

  - OAuth2 - You'll have a Bearer Token that will authenticate your API request

    ```
    var req.auth = "Bearer " + botToken||userToken;
    ```

Practice

- Download/Install Postman
- Register for a free openweathermap.org API key
- Request the weather for your home city
- Post the results in the What is an API Spark Room
- Download a good text editor if you don't have one (VSCode, Notepad++ are two good ones)