



# Webalacrity Op 1 No 1

## Programmer's Journal

---

Recipe Manager



# Programming Tools (Windows Implementation)

---

- Angular (v. 14.0.3) - installed globally with NPM
- NodeJS (v. 16.15.1) - backend JavaScript environment
- Elasticsearch (8.3.1) - backend database + NoSQL query language
- Kibana (8.3.1) - Elasticsearch data viewer and manager
- NPM Module Extras
  - Bootstrap (5.2.0)
  - jQuery (3.6.0)
  - Express (4.18.1)
  - Elasticsearch Client (8.2.1)

# Prerequisites

---

- JavaScript proficiency
- Proficiency with a source code editor/compiler (e.g. Visual Studio Code)
- JSON and JavaScript Object format familiarity
- Some acquaintance with Angular
- Basic knowledge of Visual Studio Code editor
- Concepts of source control
- Basic knowledge of NPM
- Basic knowledge of what an API is
- Browser debugger familiarity
- Basic knowledge of GitHub and Git

# Tool Installations

---

- NPM via NodeJS
- Visual Studio Code
- Git Bash
- Git for Windows
- Angular
- Angular Material
- Bootstrap
- jQuery
- Elasticsearch
- Kibana

# NodeJS

---

The installation of NodeJS will automatically install NPM. NPM is used to install NodeJS modules within Angular and NodeJS backend.

Installation Website

[NodeJS](#)

# Visual Studio Code

---

Download:

[Visual Studio Code](#)

This is a wonderful source editor and compiler for many different programming languages. One can use this to debug and step through code on the backend or frontend.

# Git Downloads

---

## [Git Website](#)

Git is the source control tool to keep track of all programming changes. The host used here is GitHub. Very useful in creating branches to source, checking out previous versions and adding tags to the Git source tree.

All the code for this project will be available online at GitHub which can be downloaded using git commands.

# GitHub

---

Location for source code changes.

- Initialize project with a git repository.
- Initial commit of start of project.
- Link project to GitHub so changes can be push up to offsite repo.
- Clone GitHub repo to download source code locally.



# Angular Global Install (NPM)

---

Install Angular globally using NPM.

Command line:

```
npm uninstall -g @angular/cli
```

```
npm install -g @angular/cli@latest
```



# Webalacrity Op 1 No 2

## Programmer's Journal



Recipe Manager



# Initialize a new Angular Project

---

Create a new Recipes project

```
ng new recipes
```

# Angular Material (angular cli)

---

Installation within Angular CLI

Used for GUI enhancements, e.g. Drag and Drop interface.

```
ng add @angular/material
```

# Bootstrap (NPM)

---

Installation:

```
npm i bootstrap@5.2.0-beta1 --save
```

```
npm i bootstrap --save (to get the latest stable)
```

Front-end toolkit which can be used in Angular by installing the module using NPM. Provides very attractive user interface widgets.

Add to Angular.json configuration

```
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.min.css",  
  "src/styles.css"  
],
```

# jQuery (NPM)

---

Installation:

```
npm i jquery --save
```

Add to Angular.json configuration

```
"scripts": [  
  "node_modules/jquery/dist/jquery.min.js"  
]
```

Used for accessing some legacy functionality provided by JavaScript jQuery. Typically accessing the jQuery object is via the \$ symbol.

# Checking the IP address of Computer

---

From the command line

```
ipconfig
```

```
Wireless LAN adapter Wi-Fi:
```

```
Connection-specific DNS Suffix  . :  
Link-local IPv6 Address . . . . . : fe80::5c4f:616c:42b2:7958%13  
IPv4 Address. . . . . : 192.168.1.203  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.1.1
```

The IPv4 Address gives you the IP address we will use to create a local server to handle security and access to database. It is also the address to use to configure elasticsearch and kibana. In this case IP = 192.168.1.203



# Webalacrity Op 1 No 3

## Programmer's Journal



Recipe Manager





# Elasticsearch & Elasticsearch Client (NPM)

---

Download site:

[Elasticsearch](#) (backend installation from zip file)

Installation into Angular via NPM:

```
npm install @elastic/elasticsearch --save
```

Used at the backend to access Elasticsearch database using queries.

# Kibana

---

Download site:

[Kibana](#) (backend installation from zip file)

An excellent online Elasticsearch manager of all the documents and indices stored in Elasticsearch. Can do queries on-the-fly. Allows you to easily create new indices of documents and experiment with data.

Setup Kibana security from Elasticsearch setup

# Security Setup for Kibana

---

```
kibana-setup --enrollment-token <token>
```

The token can be found after the output of the first execution of elasticsearch program from command line. You find the token in this line of the elasticsearch terminal output. This is generated randomly in elasticsearch.

-> Configure Kibana to use this cluster:

\* Run Kibana and click the configuration link in the terminal when Kibana starts.

\* Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):

```
eyJ2ZXIiOiI4LjMuMCIsImFkciI6WyIxOTIuMTY4LjEuMjAzOjkyMDAiXSwiZmdyIjoibWVwOTM0MjkwNmFjYzI0MzMlMjI2ZTNkNzEzZDRiYzRjZDQ4ZGVmODBmZjBiMWYxZmE4OWFkZDI0NGRhODBkOSIsImtleSI6IjJsVVBzSUVCR1p4RXNMZmlCcU
```

# First Launch of Kibana

---

The first time you launch Kibana, you will required put enter credentials.

User Name: elastic

PW: QvCfZ4hqE7DJ+EcKQj4e (randomly output from elasticsearch message on first launch)

# Elasticsearch Cluster

---

See this location of definitions of clusters, nodes, replicas, shards, etc.

## Elasticsearch Cluster

- Cluster = 3 or more nodes
- Node: is a single ES instance
- Index: collection of ES documents distributed among nodes
- Document: analogous to a record instance in a relational database
- Shard: index made up of shards each containing parts of index. Each index contains one primary shard and one replica
- Replica: used to back-up documents
- All data is defined in JSON format in Elasticsearch
- Elasticsearch is a NoSQL database and uses a special query language to access the data (also defined in JSON format)

# Checking the status of Elasticsearch Shards

---

[https://192.168.1.203:9200/\\_cluster/health?pretty](https://192.168.1.203:9200/_cluster/health?pretty)

[https://192.168.1.203:9200/\\_cat/shards](https://192.168.1.203:9200/_cat/shards)

In order to move the status from yellow to green you need replicas of the data which means the database is clustered on different computers (or nodes).

The IP address 192.168.1.203 is defined on your particular computer and may differ from the above example. Use ipconfig in DOS command window to find your address.

# JSON

---

JSON validator:

[JF Best JSON Formatter](#)

JSON is used as the primary data format for configuration and data storage.  
JSON is the text format for JavaScript Objects.



# Webalacrity Op 1 No 4

## Programmer's Journal



Recipe Manager





# Initializing Git Repo for Project

---

Navigate to your project folder and from the command line invoke the following command. This will create a hidden .git folder in your project folder and will represent your code changes repository. After initialization open the Git-GUI program and open your project folder to see your repository status.

```
git init
```

# Recipe Manager Web App Features

---

- Recipe Database and Menu
- Recipe Editor
- In Stock Inventory Tracker
- Shopping List based on menu selection
- Drag-and-Drop shopping list using inventory items
- Accessory List
- Recipe Search
- Front-end using Angular
- Back-end using Elasticsearch as database and query
- REST API interface to data
- Accessible from mobile devices such as phones and iPad
- Accessible on home WiFi using IP of computer
- Potentially accessible from a public IP address and domain URL
- Export data to JSON files for backup
- Import data from JSON files to initialize database

# Defining the Data Model - Recipe

---

- ❑ identifier (keyword)
- ❑ name (text)
- ❑ mealType (text)
- ❑ linkName (text)
- ❑ notes (text)
- ❑ ingredients (array)
  - ❑ item (number)
  - ❑ description (text)
- ❑ instructions (array)
  - ❑ step (number)
  - ❑ amount (text)

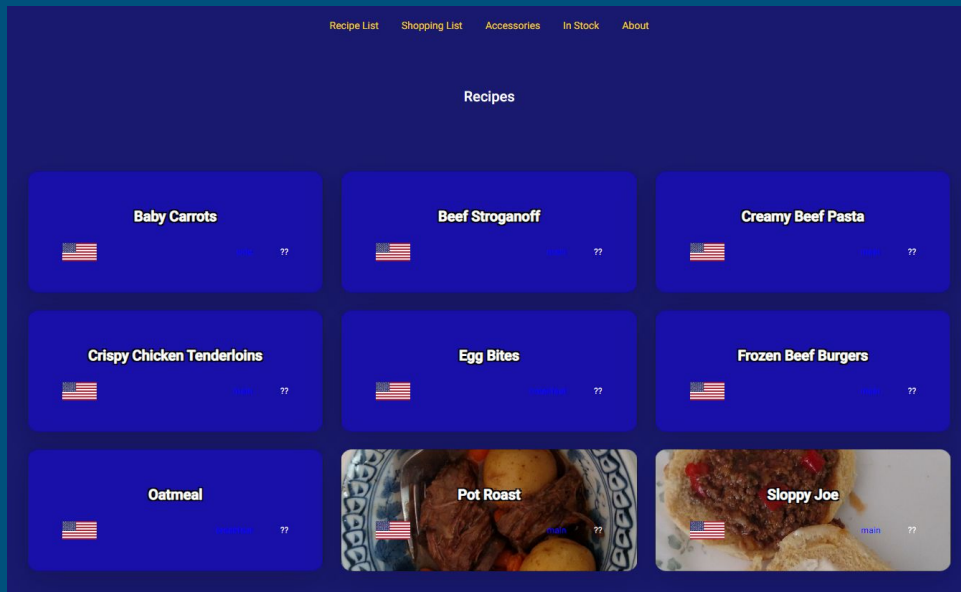
# Defining the Data Model - Inventory

---

- ❑ identifier (keyword)
- ❑ category (text)
- ❑ name (text)
- ❑ instock (boolean)

# Design Concept

## Recipe List Choices (landing page):




# Design Concept

## Recipe Detail Page:

[Recipe List](#) [Shopping List](#) [Accessories](#) [In Stock](#) [About](#)

### Egg Bites



You can get creative on this as far as ingredients goes. Chunks of bacon, ham, sausage or any other chopped meat can be added. You can add paprika or any other spice. The aluminum foil wrapping is optional. I looked at 2 recipes one used it and one didn't. I suppose the foil will keep out some extra moisture. You can try it both ways.

Source of Recipe:  
[How to Make Egg Bites in the Instant Pot | Starbucks Egg Bites](#)

#### Ingredients

Item	Amount
eggs	4
cottage cheese	1/4 cup
half/half milk	2 tbsp
paprika	sprinkle
garlic salt	sprinkle
olive oil	spray
onion diced	sprinkle
cheddar cheese	1/3 cup

#### Recipe Instructions

1. In a mixing bowl, crack open 4 large eggs (per egg bite mold).
2. Add 1/4 cup of whipping cream or cottage cheese or milk. I typically use the low-fat cottage cheese. The cream will make it more decadent and you may have to do more exercise.
3. Add 2 tablespoons of half/half to the mixture for typically sized large eggs
4. Sprinkle in a bit of cheddar cheese or any other cheese (optional).

# Design Concept

## Shopping List Page (drag and drop)

[Recipe List](#) [Shopping List](#) [Accessories](#) [In Stock](#) [About](#)

[Print Shopping List](#) [Refresh List](#) [Clear Shopping List](#)

☐ Spices

☐ Meat

☐ Condiments

☒ Vegetables

☒ Fruit

☐ Dairy

☐ Broth

☐ Oils

☐ Pasta

☐ Grains

☐ Confections

☐ Drinks

☐ Vitamins and Herbs

☐ Pastries

☐ Snacks

☐ Garden

☐ Medication

☐ Other

Potential Needed Items

baby bella mushrooms

baby carrots

baby potatoes

baked beans

cherry-tomatoes

Shopping List

bell peppers

grapes

sweet potato fries

avacados

# Design Concept

## Inventory Checklist Page(in/out of stock items)

[Recipe List](#) [Shopping List](#) [Accessories](#) [In Stock](#) [About](#)

### In Stock

#### Spices

- ☒ basil
- ☒ black pepper
- ☒ cayenne
- ☒ celery salt
- ☒ cinnamon (ground)
- ☒ garlic powder
- ☒ garlic salt
- ☒ garlic minced (can)
- ☒ garlic minced (bottle)
- ☒ ginger (ground)
- ☒ italian seasoning
- ☒ paprika
- ☒ parsley (flat leaf)
- ☒ rosemary
- ☒ salt
- ☒ thyme (ground)

#### Meat

- ☐ beef chuck
- ☐ chicken tenderloins
- ☐ frozen beef patties
- ☐ ground beef
- ☐ pork sausages



# Design Concept

## Accessories & Conversions Page

Recipe List	Shopping List	Accessories	In Stock	About
<h3>Accessories</h3> <p><b>Contained:</b> Set of glass containers. Bought at King Soopers.</p> <ul style="list-style-type: none"><li>• Microwavesable but will get hot</li><li>• Oven safe up to 450 degrees</li><li>• Dishwasher safe</li><li>• Freezer Safe</li></ul> <p>Silicon Sling for the Instant Pot (Walmart)</p> <p>Silicon Egg Bite Mold for the Instant Pot (Walmart)</p> <p>Green Bowl which weighs 6.5 oz - used for weighing ingredients</p> <p>Digital Kitchen Scale (Dash of That)</p> <h3>Ingredient Brand Names</h3> <ul style="list-style-type: none"><li>• Aiello's Spaghetti Sauce</li><li>• Barilla Mini Farfalle Bowtie Pasta (1 lb)</li><li>• Blue Elephant Royal Thai Cuisine Coconut Cream (Sprouts)</li><li>• Diced Mirepoix Mix (Onions, Carrots, Celery) (Walmart)</li><li>• Gold Meadow Viva Low-fat Cottage Cheese (Walmart)</li><li>• Goodcook Touch Veggie Dicer</li><li>• Kroger Non-stick Extra Virgin Olive Oil Cooking Spray</li><li>• Kroger Garlic Minced</li><li>• Kroger Garlic Salt</li><li>• Kroger Italian Seasoning</li><li>• Kroger Sliced Mushrooms Baby Bella</li><li>• Kroger Marinara Pasta Sauce (24 oz)</li><li>• Ore-Ida Sweet Frozen Potato Crinkle Cut Fries (French Fried Sweet Potatoes)</li><li>• Pacific Foods Organic Beef Broth</li><li>• Pompeian Extra Virgin Olive Oil</li><li>• Quaker Oats Old Fashioned (King Soopers or Walmart)</li><li>• Simple Truth Frozen Sausage Links (King Soopers)</li></ul>				



# Webalacrity Op 1 No 5

## Programmer's Journal



Recipe Manager



# Recipe Index Schema (settings)

---

```
PUT recipe
{
  "settings": {
    "analysis": {
      "analyzer": {
        "comma_delimited_analyzer" : {
          "type": "custom",
          "tokenizer": "comma_delimited_tokenizer",
          "filter": [
            "lowercase",
            "asciifolding"
          ]
        }
      },
      "tokenizer": {
        "comma_delimited_tokenizer" : {
          "type": "pattern",
          "pattern": "(, *)"
        }
      }
    }
  }
}
```

# Recipe Index Schema (mapping)

```
get recipe/_mapping
{
  "properties": {
    "ingredients": {
      "type": "text",
      "analyzer": "keyword"
    },
    "name": {
      "type": "text",
      "analyzer": "keyword",
      "fields": {
        "similarity": {
          "type": "text",
          "analyzer": "standard"
        }
      }
    },
    "recipe_type": {
      "type": "text",
      "analyzer": "keyword"
    },
    "cooktime": {
      "type": "text",
      "analyzer": "keyword"
    },
    "dishsource": {
      "type": "text",
      "analyzer": "keyword"
    },
    "notes": {
      "type": "text",
      "analyzer": "keyword"
    },
    "ingredients2": {
      "type": "text",
      "analyzer": "keyword",
      "fields": {
        "similarity": {
          "type": "text",
          "analyzer": "standard"
        }
      }
    },
    "amount": {
      "type": "text",
      "analyzer": "keyword"
    },
    "instructions": {
      "type": "text",
      "analyzer": "keyword",
      "fields": {
        "step": {
          "type": "text",
          "analyzer": "keyword"
        },
        "description": {
          "type": "text",
          "analyzer": "keyword"
        }
      }
    }
  }
}
```



# Webalacrity Op 1 No 6

## Programmer's Journal

---

Recipe Manager



# Adding a new repo to GitHub

---

```
git remote add origin <your github account>/<project name>.git  
git push -u origin master
```

This will store some information of the GitHub site into your `.git/config` file automatically and tie your local repo to the GitHub site. Not needed by user unless you are creating your own GitHub project. Once done, you can push your committed changes to GitHub or allow others to clone your project.

# Cloning a project from GitHub

---

In a command line, go to a directory you want to contain a GitHub project. Use this command to clone the project files.

```
git clone <project URL>
```

# Turning your angular app into a server

---

In order for your web application to be available on other devices on your WiFi network you will need to turn it into a server. This will create a true NodeJS backend to communicate to using REST API commands that you define. These commands will allow data to be retrieved from the Elasticsearch database publicly on the network. You need to install the NPM package in your project:

```
npm install express --save
```



# Creating a real Node service call

---

In order to create a backend server (which has access to database) for this project you will need to create a NodeJS service which you can call from the command line. All URL calls to your app which are not caught by Angular will be routed to the Backend server code. The essence of the server code (server.js) to do this is:

```
app.use('/api', api);
app.use('/api-import', apiImport);

// Catch all routes
app.get('*', (req, res) => {
  res.sendFile(path.join(__dirname, 'dist/recipes/index.html'))
});

const port = process.env.PORT || 98;

const ip = process.env.IP || '192.168.1.203';

app.listen(port, ip, (req, res) => {
  console.log(`Running on ip: ${ip} port: ${port}`);
});
```

# NodeJS backend code

---

The actual NodeJS backend code which will intercept your REST API calls will have the following form:

```
'use strict'

const express = require('express');
const fs = require('fs');
const router = express.Router();

// These two statements allow parsing of body correctly when
// getting a POST request (otherwise req.body may equal undefined)
router.use(express.json());
router.use(express.urlencoded({ extended: true }));

// Get all recipes
router.get('/recipes/all-recipes', (req, res) => {
  // code to access database goes here. This intercepts a specific REST API call to retrieve all recipes and return to client
});

module.exports = router;
```