*Alexander Holm*

*N9918205*

*Cloud Computing*

Mashup/ Docker ProjectAssessment 1

Table of Contents

[Introduction 1](#_Toc524952101)

[Mashup use cases and services 2](#_Toc524952102)

[Case 1: 2](#_Toc524952103)

[Case 2: 3](#_Toc524952104)

[Case 3: 4](#_Toc524952105)

[Technical description of the application. 5](#_Toc524952106)

[Issues Encountered 7](#_Toc524952107)

[Discussion of the use of Docker 8](#_Toc524952108)

[Testing and limitations 9](#_Toc524952109)

[Error Handling 9](#_Toc524952110)

[Possible extensions 11](#_Toc524952111)

[Appendix: 11](#_Toc524952112)

# Introduction

I keep telling myself that I will eat better but I never do.

MindfulMeals will hope to help users make better dinner decisions. Using Food2Fork, Edamam and Google Charts, users can search for recipes using key ingredients and recipe titles, from here they can explore the results from their query and be presented with information about the meal to guide users to a healthy choice.

Users provide a search query, either a key ingredient or specific title of a recipe, the results include a title, image and a source URL. The results are then individually sent to Edamam, which provides the ingredients and nutritional analysis of that specific meal. This provides the user with the ingredients, serving size and a bar graph of the Recommended Daily Intake as a percent.

**Food2Fork**

<https://www.food2fork.com/api/search>

Food2Fork provides two callable methods, one for a general search and one that provides some details a specific recipe. They have various parameters for these methods. MindfulMeals uses the search. to fetch a title, image and source.

Search has 4 parameters;

The API key, the query, sort by (rating or trend) and page for returning extra results.

The API key and query are required for the call to return a non-error.

**Edamam**

<https://api.edamam.com/search>

The second API is Edamam’s search API, which offers a full analysis of a recipe, in real time.

The information is delivered as a response from a GET request. This request can specify different parameters; calories, health tags such as peanut free or vegan, time taken to cook and a recipe exclusion list for picky eaters.

The response provides a full recipe analysis of the recipe title’s provided by food2fork, which includes, ingredients, fat, carbs, protein in each serving.

**Google Charts**

<https://developers.google.com/chart/>

Google Chart API is a web service that create graphs from data. Many forms of graphs can be made.

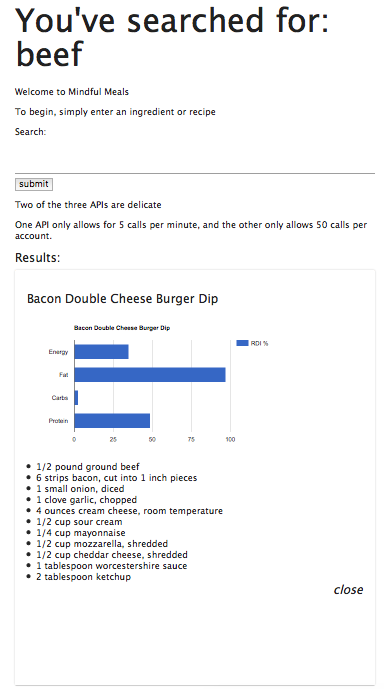
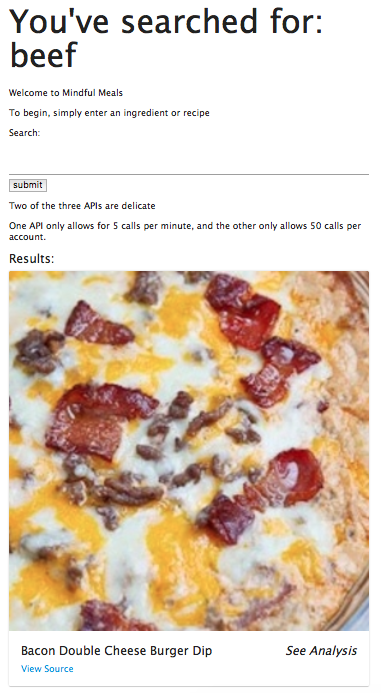
MindfulMeals will be using a bar chart to display the recommended daily intake of the recipe.

Mashup use cases and services

## Case 1:

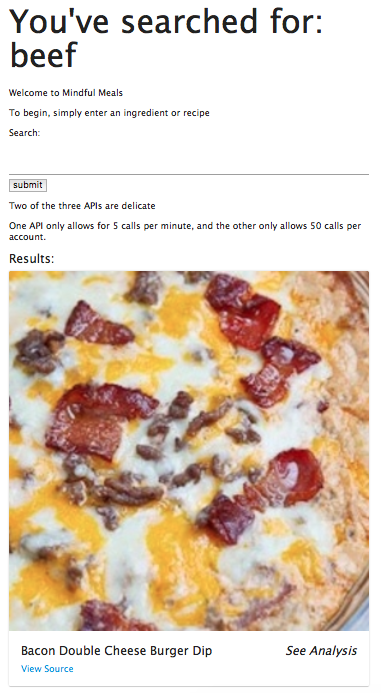
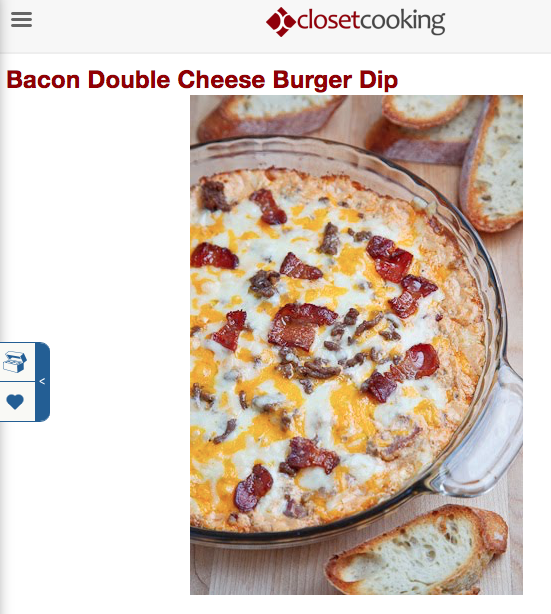
**As a user, I wish to search for recipes easily that appeal to the flavour I am craving.**

This user from the index can enter a query for beef, click the Search button and will be given a list of recipes, and their ingredients.

****This use case uses Food2Fork, Edamam

Case 2:  
 **As a user, I wish to be able to find the original creator of the recipe, so that I may also see when else they have done.**

The user makes a query, and decides they would like to visit the recipe authors’ webpage.  
At the bottom left of the results card, there is a link *“View Source”* which will redirect the user to the source of the recipe

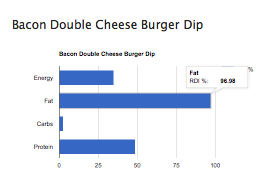
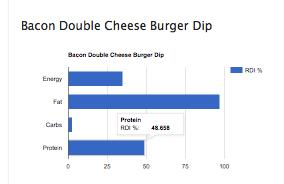
 

This use case utilised only the Food2Fork API.

## Case 3:

**As a user, I’d like to closely analyse the food I am preparing, so I can reduce my fat intake and improve my protein.**

When the user finds a meal they find appealing, they can click on *“View Analysis”* for a list of the ingredients as well as a graph showing them their recommended daily intake of energy, fat, carbs and protein.



The user decides to opt for another meal is this one has over 90% of their daily fat intake.

This use case utilises Food2Fork, Edamam and Google Charts

­Technical description of the application.

This project has been pushed my limits of code knowledge and taught me many things, there is much more I wish I could have learnt or implemented. Particularly, creating some separation from client and server. This thing is sluggish, my hand is oozing plasma, I can’t get it cleaned and dressed and I have a report to write while there is a small storm in my head going on.

So, I’m writing this very casually before 1pm because after that, my university work day is over.

When the user makes a query, the Client sends a request using the ‘request-promise’ package to the Food2Fork API using a GET request. This responds with unfiltered boilerplate as visible below. This response is filtered into:The title is taken from this filtered response and sent to Edamam for further information gathering. The response for this recipe is found on the next page:



This massive chunk of data is filtered into: 

This is the final set of information needed to generate the page.

Energy, Fat, Carbs and Protein are sent to Google’s chart for displaying the RDI in an easy to read format. All the responses are cleaned by the bodyParser package which is native to express.

# Issues Encountered

Two APIs had persisting pains.

Food2Fork only gives 50 calls per account, meaning that at least twice a day, I would have to create a new account, fortunately they have no email verification. I overcame this issue by creating new accounts under non-existent emails.

Edamam proved to be friendlier to work with as they had a 5 call per minute restriction for free accounts. Overcoming this issue meant that I had to write my code two or three steps ahead of what I was trying to do, so if it worked I could see what error to fix next.

Another issue that came from this API was making small changes in code had a 1 minute cooldown.

Initially, this was one big page of mom’s spaghetti, so I got sweaty and made some changes to make my code express ready.

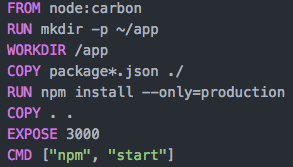
Other issues I encountered were growing pains learning about Amazon Web Services, Docker and relearning parts of JavaScript.

Discussion of the use of Docker

, configuration choices. You may include

Docker is awesome.

Learning to use Docker has been so much fun. Docker has made it so much easier to deploy and run the application.



This is the Dockerfile used to build and run MindfulMeals.

FROM NODE: CARBON  
Builds the image from NODE.JS release v8.9.0

RUN mkdir -p ~/app  
Runs the command in an shell, the command makes a directory under the root called app. The ‘-p’ argument specifies the path.

WORKDIR /app  
Assigns the working directory, which was made in the line prior.

COPY package\*.json ./  
This copies the package.json file. The \* allows for package-lock.json to be copied if it exists.  
  
RUN instruction will execute the command “npm install –only=production” which installs the dependencies that are only used for production.  
  
COPY . .  
Will bundle the app’s source code into the Docker image.

EXPOSE 3000  
This command binds the app to port 3000

CMD [“npm”, “start”]  
This defines the runtime. Running npm and start which in turn, starts the server.

Testing and limitations

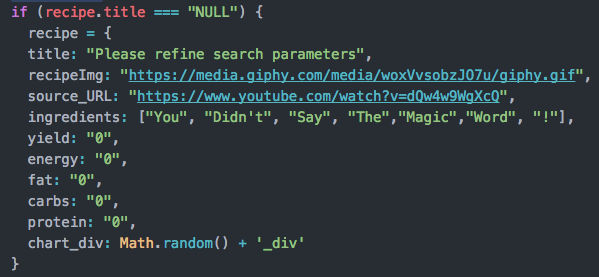
## Error Handling

There are a few tricks to handle errors.

If the user puts in a non-valid query like ‘adolf’, Food2Fork’s response will be undefined.

If the response is not undefined, the function will run as normal, otherwise, this indicates that the response is not valid and assigns the recipe to “NULL”

Later on when the recipe value is passed to the Edamam API, it checks the title. If the title is “NULL”, it creates a recipe object that indicates an error occurred.



If the recipe title provided by Food2Fork causes an issue in the Edamam. An example of this is “How to make Gnocchi like an Italian grandmother”

To handle this error, I used a try catch. If everything works, the recipe returns complete, if there is an error, it will populate the rest of the recipe with warnings and the user can use the source URL to view the ingredients.



Possible extensions

Extending Use of Edamam API  
I hate making trivial stuff for assignments. After this assignment has been handed in, I am going to change some of settings to allow me to search for vegan only. I tried to implement this towards the end but found PUG to give grief with checkboxes and decided it wasn’t worth the time to troubleshoot

Front End Framework  
The app is very simple in appearance and can be more appealing to the eye. With some more time and patience, this webpage could have a better homepage.

Database Usage  
Initially, I was using MySQL to persist the data. I could only use up to 500mb for free and decided that it wouldn’t be helpful in the long run. I also tried to start using MongoDB but due to some complications, it was scrapped as other solutions were available.

Edamam Cooldown  
I wanted to implement a countdown timer that showed the time needed for the API calls to cooldown but found this method of handling errors too difficult at the stage I wanted to implement it.  
If academic integrity was not involved, I would have used a simple timer from GITHUB

Appendix:

User Guide:  
The App is very intuitive and provides clear instructions on the home page.  
All the user needs to do, is simple enter an ingredient or recipe and click the sumbit button. This page is visible when the user makes a search so they don’t have to navigate back to the home page to query.

