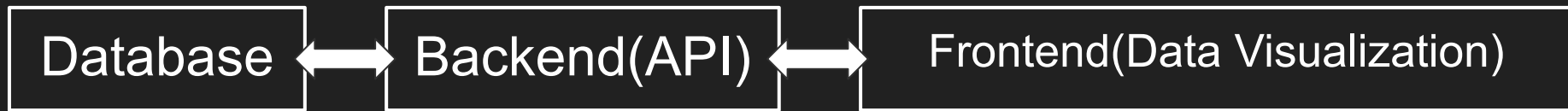


Fast Food Dieter

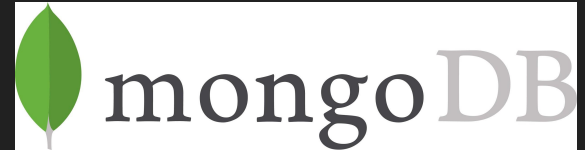
Your new diet guide solution

Motivation

Architecture



Datasource and Data Acquiring Methodology



Database

- The simplicity of NoSQL gives flexibility in table structure which reduces time in designing the database.

fastfood.burger

COLLECTION SIZE: 928B TOTAL DOCUMENTS: 8 INDEXES TOTAL SIZE: 36KB

[Find](#) [Indexes](#) [Aggregation](#)

INSERT DOCUMENT

FILTER

{ "filter": "example" }

Find

Reset

QUERY RESULTS 1-8 OF 8

```
_id: ObjectId("5de404c61c9d440000173dea")
name: "kfc chicken burger"
calories: 530
saturated_fat: 27
brand: "kfc"
```

```
_id: ObjectId("5de405fc1c9d440000173dec")
name: "kfc fish burger"
calories: 470
saturated_fat: 22
brand: "kfc"
```

API

- We use express to create a powerful RESTful API.
Simple and efficient.

```
1  require("dotenv").config();
2  const express = require('express');
3  const MongoClient = require('mongodb').MongoClient;
4  const router = express.Router();
5
6  router.get('/test-food', async (req, res) => {
7    const tmp = await loadFoods("foods");
8    res.send(await tmp.find({}).toArray());
9  });
10
11 router.get('/salads', async (req, res) => {
12   const collection = await loadFoods("salad");
13   res.send(await collection.find({}).toArray());
14 });
15
16 router.get('/chickens', async (req, res) => {
17   const collection = await loadFoods("chicken");
18   res.send(await collection.find({}).toArray());
19 });
20
21 router.get('/rice', async (req, res) => {
22   const collection = await loadFoods("rice");
23   res.send(await collection.find({}).toArray());
24 });
25
26 router.get('/burgers', async (req, res) => {
27   const collection = await loadFoods("burger");
28   res.send(await collection.find({}).toArray());
29 });
30
```

Data Visualization

- React is use to display datas in graph to show calories per each food.

