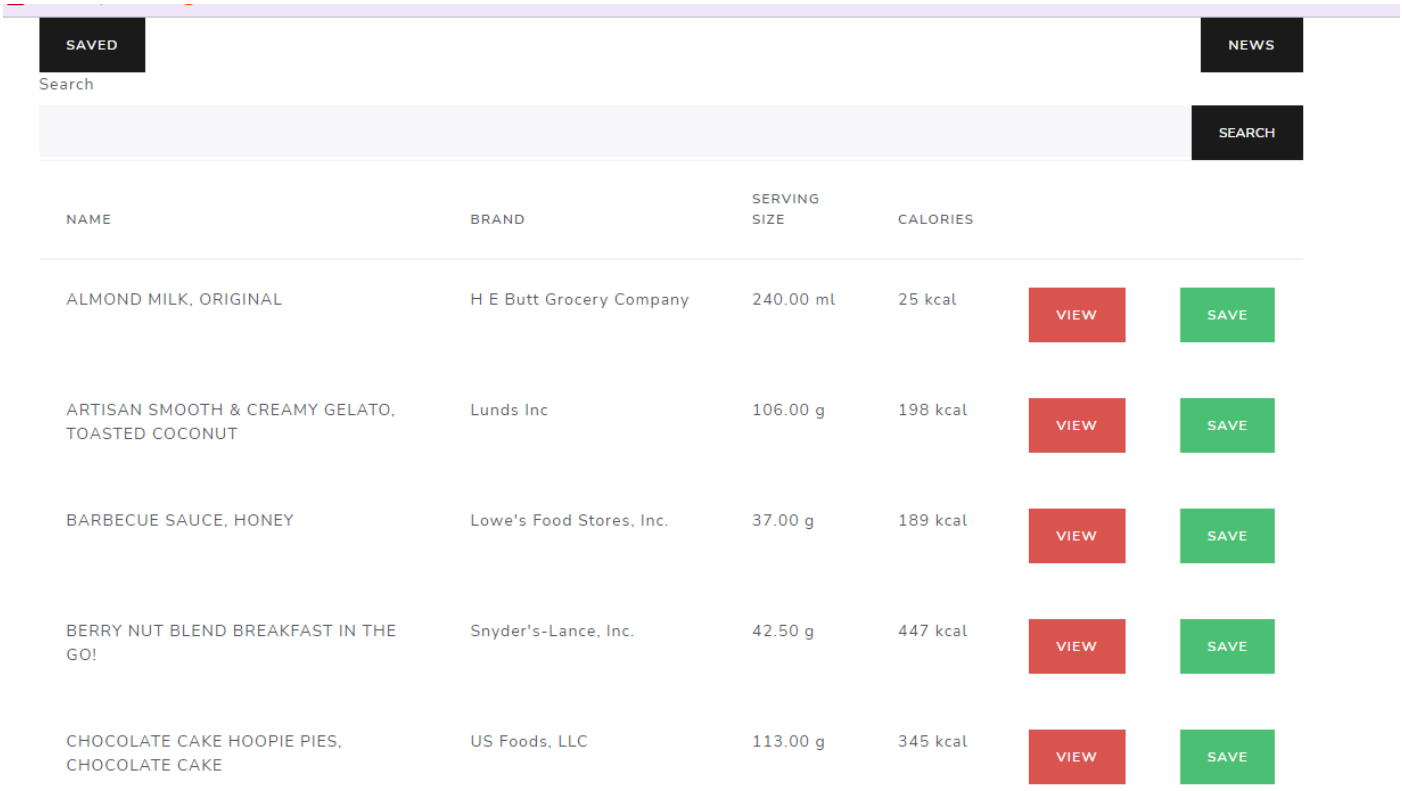
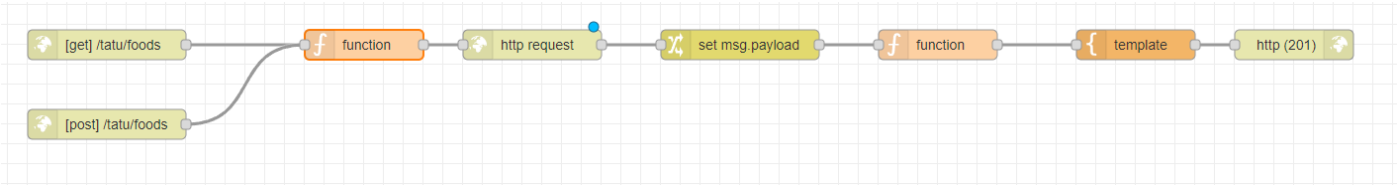
Tatu Bogdan – IIOTCA Project

First page:





I started by creating a foods page:

Where I'd have a list of foods that I got from an external api ("USDA FoodData Central") with a GET request:

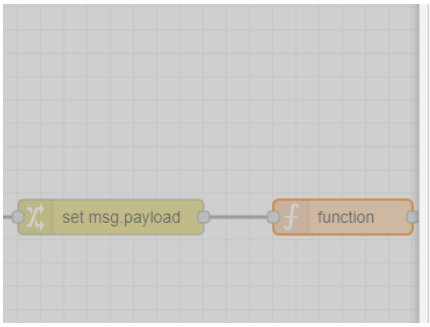
I needed to pass in a few query-string parameters, so I put a function node before that set the message payload to:

```
const { search } = msg.payload
msg.payload = {
    api_key: "U90ctjxNwICMOKnJnyqB2kDuKgpqL2dG7vBB03lZ",
    dataType: "Branded",
    generalSearchInput: search,
    page: 0,
    pageSize: 10
}

return msg;
```

The api also allows for the filtering of the results.

I then changed the data of the payload to something more useful and readable:



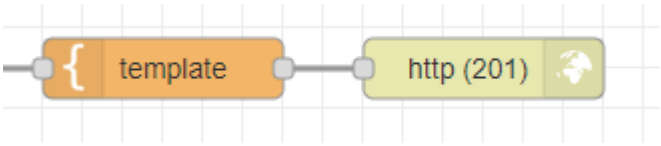

```
const cringe = msg.payload.map(food => {
    const { fdcId, description, brandOwner, foodNutrients, servingSize, servingSizeUnit } = food

    return {
        id: fdcId,
        name: description,
        brand: brandOwner,
        calories: foodNutrients.find(nutrient => nutrient.nutrientId === 1008).value,
        serving: `${servingSize.toFixed(2)} ${servingSizeUnit}`
    }
});

msg.payload = cringe

return msg
```

And made the actual HTML page to present the data:




```
<div class="container">
    <div class="d-flex justify-content-between">
        <div><a href="/tatu/saved" class="btn btn-primary">saved</a></div>
        <div><a href="/tatu/news" class="btn btn-primary">news</a></div>
    </div>
    <form method="POST" action="/tatu/foods">
            <label for="search">Search</label>
        <div class="d-flex">
            <input class="form-control" id="search" name="search">
            <button type="submit" class="btn btn-primary">Search</button>
        </div>
    </form>
</div>
<div class="container">
    <table class="table table-hover">
        <thead>
            <tr>
                <th scope="col">Name</th>
                <th scope="col">Brand</th>
                <th scope="col">Serving Size</th>
                <th scope="col">Calories</th>
                <th scope="col"></th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody>
            {{#payload}}
            <tr class="table-secondary" onclick="">
                <td>{{name}}</td>
                <td>{{brand}}</td>
                <td>{{serving}}</td>
                <td>{{calories}} kcal</td>
                <td><a href="/tatu/food/{{id}}" class="btn btn-danger">view</a></td>
                <td><a href="/tatu/saved/{{id}}?name={{name}}&brand={{brand}}&serving={{serving}}&calories={{calories}}" class="btn btn-success">save</a></td>
            </tr>
            {{/payload}}
        </tbody>
    </table>
</div>
```

Where I have a few links that go to different pages that will be explained later and a POST request form for the search bar to go to the same url, and have the payload.search set to the input, so that I can use it for filtering.

Second page:

BERRY NUT BLEND BREAKFAST IN THE GO!

SNYDER'S-LANCE, INC.

NUTRIENT	AMOUNT	UNIT
Energy	447	kcal
Protein	9.41	g
Fat	21.18	g
- saturated	4.71	g
- trans	0	g
- cholesterol	0	mg
Carbs	58.82	g
- fiber	7.1	g
- sugars		

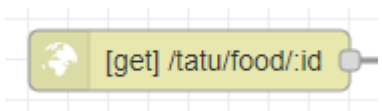
Ingredients: VANILLA GRANOLA (ROLLED OATS, BROWN SUGAR, WHEAT FLAKES, WHOLE OAT FLOUR, CANOLA OIL, CORN SYRUP, RICE FLOUR, ARTIFICIAL FLAVOR, SALT, MIXED TOCOPHEROLS [TO PRESERVE FRESHNESS], RICE EXTRACT, DISTILLED MONOGLYCERIDES, BARLEY MALT EXTRACT), COCKTAIL PEANUTS (PEANUTS, PEANUT AND/OR COTTONSEED OIL, SALT), YOGURT FLAVORED RAISINS (YOGURT COATING [SUGAR, PARTIALLY HYDROGENATED PALM KERNEL OIL, CALCIUM CARBONATE, YOGURT POWDER {CULTURED WHEY PROTEIN CONCENTRATE, CULTURED SKIM MILK, AND YOGURT CULTURE}, ARTIFICIAL COLOR, SOY LECITHIN {AN EMULSIFIER}, NATURAL FLAVOR], RAISINS, SUNFLOWER OIL, CORN SYRUP, MODIFIED STARCH, COCONUT OIL, CONFECTIONER'S GLAZE), INFUSED DRIED CRANBERRIES (SLICED CRANBERRIES, SUGAR, GLYCERIN, SUNFLOWER OIL), GLAZED WALNUTS (WALNUTS, SUGAR, CORN SYRUP, SESAME SEEDS, SALT, CANOLA OIL, NATURAL FLAVORS, SOY LECITHIN [AN EMULISIFIER], CITRIC ACID), INFUSED DRIED BLUEBERRY FLAVORED CRANBERRIES (SLICED CRANBERRIES, SUGAR, BLUEBERRY JUICE, SUNFLOWER OIL, NATURAL BLUEBERRY FLAVOR, MALIC ACID), DUSTED WITH: MALTODEXTRIN, MANNITOL, MODIFIED CELLULOSE.

BACK

The next page that I wanted to make was a page of the breakdown of a few nutrients, calories and ingredients of a given food, that can be accessed through the view button on the last page.

It's based on the same premise as the last page:

1. go to link:

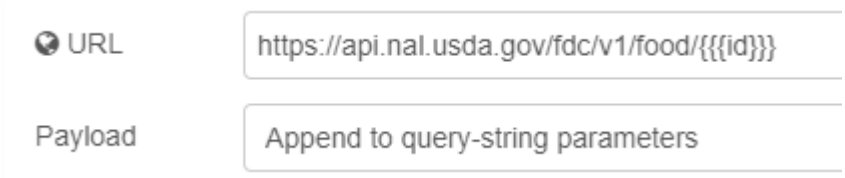


2. Set the payload to make a request:

```
msg.payload = {
    api_key: "U90ctjxNwICMOKnJnyqB2kDuKgpqL2dG7vBB03lZ",
}
msg.id = msg.req.params.id

return msg;
```

3. Make the request:



4. Process the results:

```
const findNutrient = (nutrients, id) => {
    return nutrients.find(nutrient => nutrient.nutrient.id === id)
}

const { foodNutrients, description, brandOwner, servingSize, servingSizeUnit, ingredients } = msg.payload

msg.payload = {
    energy: findNutrient(foodNutrients, 1008),
    protein: findNutrient(foodNutrients, 1003),
    fat: findNutrient(foodNutrients, 1004),
    saturated: findNutrient(foodNutrients, 1258),
    trans: findNutrient(foodNutrients, 1257),
    cholesterol: findNutrient(foodNutrients, 1253),
    carbs: findNutrient(foodNutrients, 1005),
    fiber: findNutrient(foodNutrients, 1079),
    sugar: findNutrient(foodNutrients, 200),

    name: description,
    brand: brandOwner,
    serving: `${servingSize} ${servingSizeUnit}`,
    ingredients
}

return msg
```

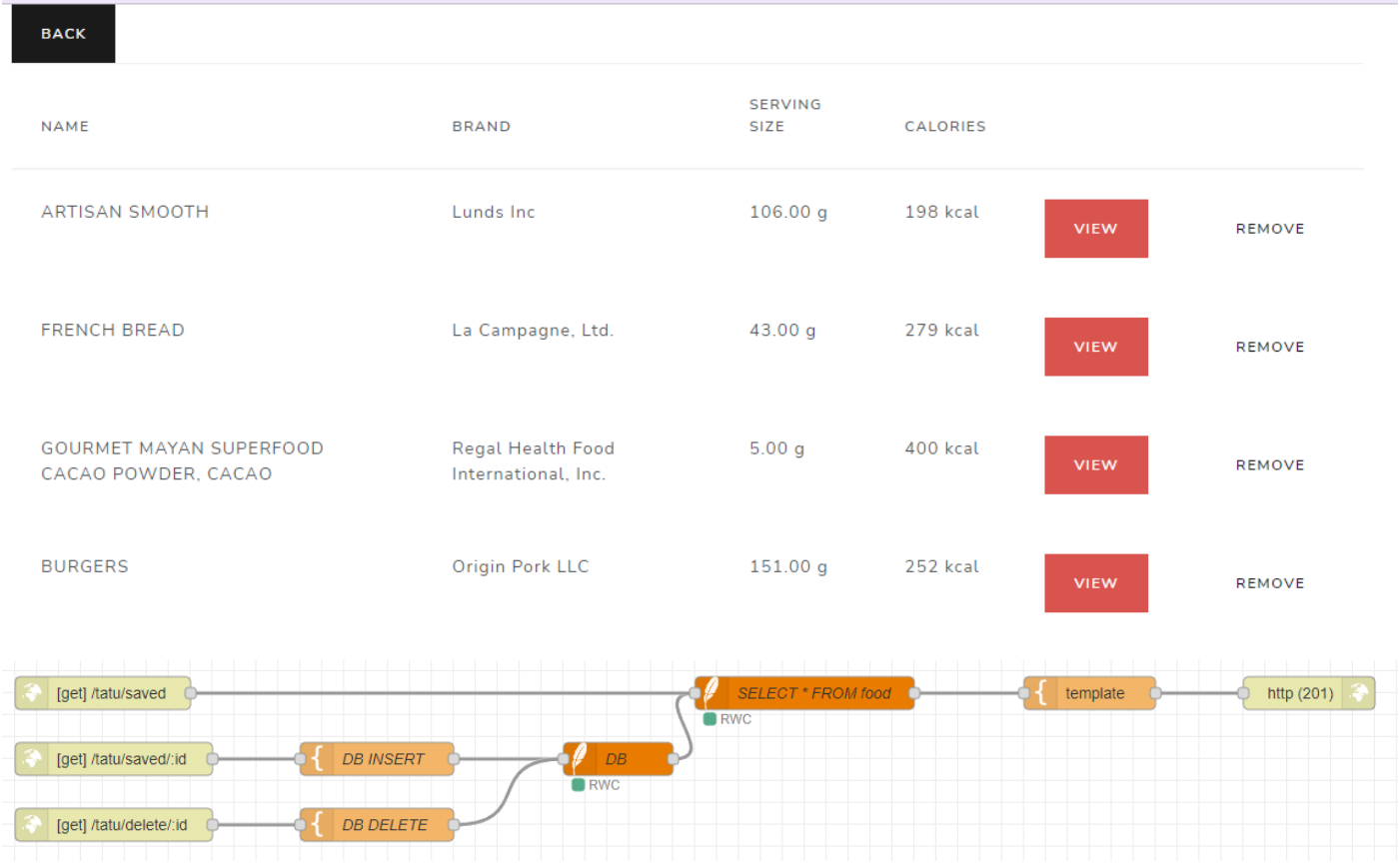
In this case I made an object that had a few nutrients out of a list of nutrients

5. Make a page to present the data:

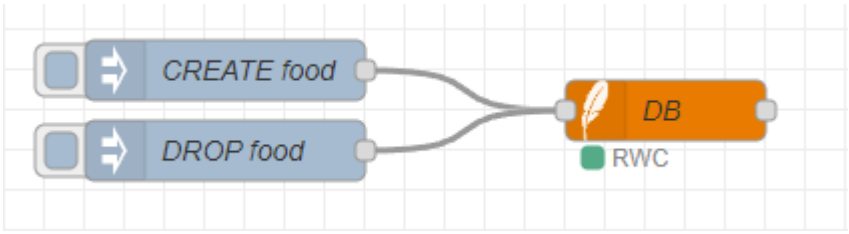
```
<body>
    {{#payload}}
    <div class="container d-flex flex-column align-items-center">
        <h1>{{name}}</h1>
        <h4>{{brand}}</h4>

        <table class="table table-hover">
            <thead>
                <tr class="table-primary">
                    <th scope="col">Nutrient</th>
                    <th scope="col">Amount</th>
                    <th scope="col">Unit</th>
                </tr>
            </thead>
            <tbody>
                <tr class="table-secondary" onclick="">
                    <td>Energy</td>
                    <td>{{energy.amount}}</td>
                    <td>{{energy.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td>Protein</td>
                    <td>{{protein.amount}}</td>
                    <td>{{protein.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td>Fat</td>
                    <td>{{fat.amount}}</td>
                    <td>{{fat.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td> - saturated</td>
                    <td>{{saturated.amount}}</td>
                    <td>{{saturated.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td> - trans</td>
                    <td>{{trans.amount}}</td>
                    <td>{{trans.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td> - cholesterol</td>
                    <td>{{cholesterol.amount}}</td>
                    <td>{{cholesterol.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td>Carbs</td>
                    <td>{{carbs.amount}}</td>
                    <td>{{carbs.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td> - fiber</td>
                    <td>{{fiber.amount}}</td>
                    <td>{{fiber.nutrient.unitName}}</td>
                </tr>
                <tr class="table-secondary" onclick="">
                    <td> - sugars</td>
                    <td>{{sugar.amount}}</td>
                    <td>{{sugar.nutrient.unitName}}</td>
                </tr>
            </tbody>
        </table>
        <div>
            Ingredients: {{ingredients}}
        </div>
        <div><a href="/tatu/foods" class="btn btn-primary">back</a></div>
    </div>
    {{/payload}}
</body>
</html>
```

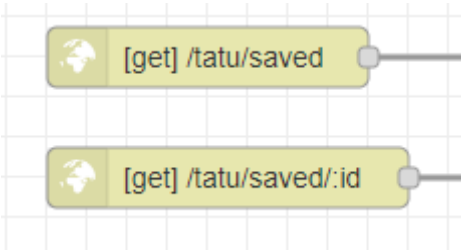
Third page:



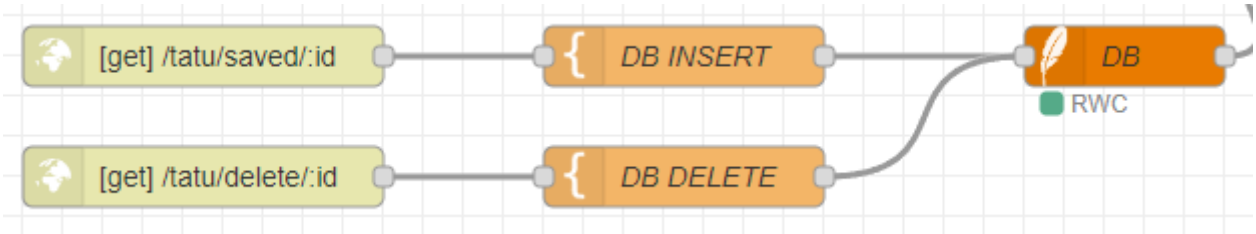
A page where I could save favorite foods to a local database:



It is accessed with the SAVED button or the SAVE buttons next to foods on the first page:



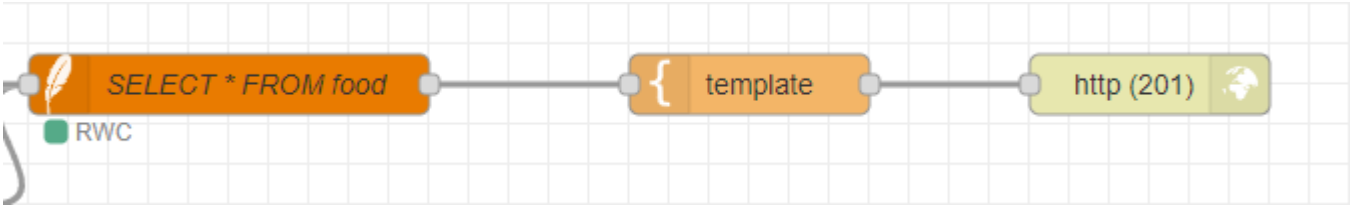
The SAVE button also inserts a new entry into the database, and the REMOVE button on this page deletes the food by its ID:



```
iplate                                                    Syntax Highlight: SQL
INSERT INTO food VALUES ('{{req.params.id}}','{{payload.name}}','{{payload.brand}}','{{payload.serving}}','{{payload.calories}}')
```

```
emplate
DELETE FROM food WHERE food.id = "{{req.params.id}}"
```

After any action everything is selected from the database:



And is passed down to the HTML template:

```
<body>
    <div class="container">
        <div><a href="/tatu/foods" class="btn btn-primary">back</a></div>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th scope="col">Name</th>
                    <th scope="col">Brand</th>
                    <th scope="col">Serving Size</th>
                    <th scope="col">Calories</th>
                    <th scope="col"></th>
                    <th scope="col"></th>
                </tr>
            </thead>
            <tbody>
                {{#payload}}
                <tr class="table-secondary">
                    <td>{{name}}</td>
                    <td>{{brand}}</td>
                    <td>{{serving}}</td>
                    <td>{{calories}} kcal</td>
                    <td><a href="/tatu/food/{{id}}" class="btn btn-danger">view</a></td>
                    <td><a href="/tatu/delete/{{id}}" class="btn btn-secondary">remove</a></td>
                </tr>
                {{/payload}}
            </tbody>
        </table>
    </div>
</body>
</html>
```

So it can be presented.

Last page:

BACK

CREAMY VEGAN STRAWBERRY MILKSHAKE (4 INGREDIENTS!)

Wed, 08 Jun 2022

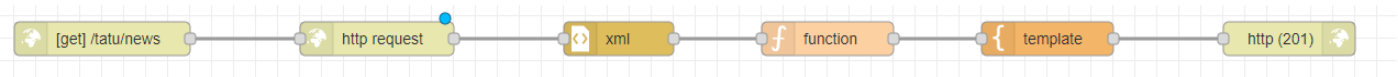
Introducing the ULTIMATE vegan strawberry milkshake — thick, creamy, classic strawberry flavor, and just 4 simple ingredients. It's maple-sweetened and made without ice cream but has ALL the fresh, rich flavor you know and love.

It's *almost* too good to be true, but you better believe it! Let's make strawberry milkshakes!

How to Make a Strawberry Milkshake Without Ice Cream

If you don't have ice cream at the moment, or you're trying to keep things naturally sweetened or avoid some of the unrecognizable ingredients in store-bought products, we have just the solution for your craving: a strawberry milkshake made without ice cream!

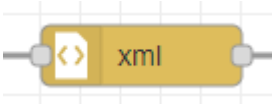
Creamy Vegan Strawberry Milkshake (4 Ingredients!) from Minimalist Baker →



I wanted to do something with RSS feeds so I found a recipe RSS that I request:

https://minimalistbaker.com/feed/

The data is passed through an xml parser node:



And finally brought to a more normal form, in a function:

```
const news = msg.payload.rss.channel[0].item.map((item, index) => {
    return {
        id: index,
        title: item.title[0],
        link: item.link[0],
        date: item.pubDate[0].match('[a-zA-z]+, [0-9]+ [a-zA-Z]+ [0-9]+'),
        description: item.description[0],
    }
})

msg.payload = news

return msg;
```

Lucky for me, the description was an HTML string, so it already had the style made.

And the data is once again presented:


```
<body>

    <div class="container">
        <div><a href="/tatu/foods" class="btn btn-primary">back</a></div>
        <div class="d-flex flex-column align-items-center">
            {{#payload}}
                <h1>
                    {{title}}
                </h1>
                <div>
                    {{date}}
                </div>
                <br />
                <br />
                <div id="{{id}}">
                    {{{description}}}
                </div>
                <br />
                <br />
                <br />
            {{/payload}}
        </div>
    </div>
</body>
</html>
```

Thank you for following my presentation, I'm a bit tired and need some sleep, but here you go:

Sorry for the delay.