howest
hogeschool

Projectdossier

EPizza project

Duncan De Vlaminck & Jonas Anne

API

Gateway: https://epizzaapi.azurewebsites.net/

Login gegevens

Login: (admin)

Username : Docent@1
Password : Docent@1

Login (Customer)

Username : Jonas@1
Password : Jonas@1

Werkverdeling

Van het begin van het project hebben we duidelijk afgesproken wie wat zou doen,
Duncan heeft hem gefocust op de Frontend en Jonas op de Backend.

Milestones

Milestone 1

- ERD af
- Design document af

Milestone 2

- Database klaar
- Begin controller uitwerking
- Views in frontend klaar

Milestone 3

- Frontend af
- Backend af

Niet behaalde targets:

- We hadden graag nog een tracking systeem ingestoken maar we streefde naar een goedwerkende basis van de applicatie.

Grootse behaalde successen

- Alles staat online.
- Mongodb cluster online gemaakt.
- We hebben een mooie basis van de applicatie (zowel frontend als backend).

Moeilijkheden

Backend

- Pizza's toevoegen met toppings in de database
- Alles online zetten met behulp van docker

Frontend

- PWA doen werken (op Netlify werkt het niet, maar op Vercel wel...)
- Reactivity
- IndexedDB

Extra's

- Zoals hierboven vermeld, alle databases staan online. Ook mongodb.

Zelfreflectie

Jonas:

Tijdens dit semester heb ik met een klein depressie gezeten waardoor het soms moeilijk was om de motivatie te vinden om te werken aan het project.
De samenwerking is vlot verlopen, in het begin was de communicatie wat moeilijk maar nu mag ik hier zeker niet over klagen.

Ducan:
Dankzij dit project heb ik mijn Vue skills kunnen verrijken en dit gaat handig zijn bij mijn stage. Ik ben blij met het resultaat van ons project, maar er zijn nog zaken dat niet afgeraakt zijn. Het was een groot project.

Pizzas

(https://epizzaapi.azurewebsites.net/pizzas)

Endpoints

- GET : https://epizzaapi.azurewebsites.net/pizzas
- POST : https://epizzaapi.azurewebsites.net/pizzas
- GET op id : https://epizzaapi.azurewebsites.net/pizzas/id
- DELETE op id : https://epizzaapi.azurewebsites.net/pizzas/id
- PUT op id : https://epizzaapi.azurewebsites.net/pizzas/id

```
// CREATE with toppings
{
    "name": "Pizza Name",
    "price": 10,
    "imgUrl": "https://img.static-rmg.be/a/food/im-
age/q75/w640/h400/1077806/pizza.jpg",
    "toppings": [{
            "name": "Pork"
        },
    ]
}
// Update zonder toppings
{
    "id": "B4040F4B-C19A-454D-9B53-CBF9612FD305",
    "name": "Tuna",
    "price": 15,
    "imgUrl": "nieuweUrl"
}
//update met toppings
{
    "id": "F684EC5C-16C0-4581-8306-2B180AF55954",
    "name": "Parma",
    "price": 8.1,
    "imgUrl": "https://cdn-catalog.pizzahut.be/im-
ages/be/20170830145618975.jpg",
    "toppings": [{
            "Id": "5F6B364B-8A29-4C91-A08C-7528F9307AAE",
            "Name": "Mozzarella"
        }
    ]
}
```

Toppings

(https://epizzaapi.azurewebsites.net/toppings)

Endpoints

- GET : https://epizzaapi.azurewebsites.net/toppings
- POST : https://epizzaapi.azurewebsites.net/toppings
- GET op id : https://epizzaapi.azurewebsites.net/toppings/id
- DELETE op id : https://epizzaapi.azurewebsites.net/toppings/id
- PUT op id : https://epizzaapi.azurewebsites.net/toppings/id

```
//--------------- toppings ---------------
// CREATE TOPPING
{
    "name": "Topping name",
    "price": 1.2,
}

// UPDATE TOPPING
{
    "id": " B4040F4B-C19A-454D-9B53-CBF9612FD305",
    "name": "Topping name",
    "price": 1.2,
}
```

PizzaReviews

(https://epizzaapi.azurewebsites.net/pizzareviews)

Endpoints

- GET : https://epizzaapi.azurewebsites.net/pizzareviews
- POST : https://epizzaapi.azurewebsites.net/pizzareviews
- GET op id : : https://epizzaapi.azurewebsites.net/pizzareviews/id
- DELETE op id : : https://epizzaapi.azurewebsites.net/pizzareviews/id
- PUT op id : https://epizzaapi.azurewebsites.net/pizzareviews/id

```
//--------------- pizzareviews ---------------
// CREATE PIZZAREVIEW
{
    "title": "string",
    "description": "string",
    "rating": 0, //op 5
    "pizzaId": "string"
}
```

Restaurants

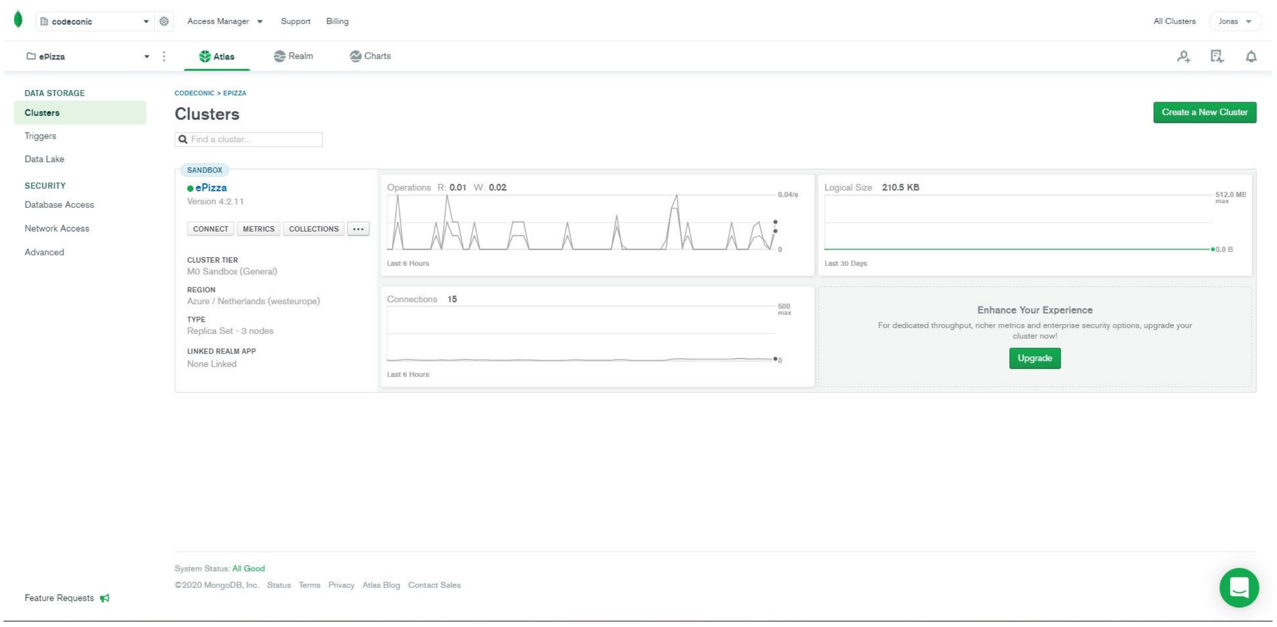
(https://epizzaapi.azurewebsites.net/restaurant)

Endpoints

- GET : https://epizzaapi.azurewebsites.net/restaurant
- POST : https://epizzaapi.azurewebsites.net/restaurant
- GET op id : https://epizzaapi.azurewebsites.net/restaurant/id
- DELETE op id : https://epizzaapi.azurewebsites.net/restaurant/id
- PUT op id : https://epizzaapi.azurewebsites.net/restaurant/id

```
//-------------- pizzareviews --------------
// CREATE PIZZAREVIEW
{
    "title": "string",
    "description": "string",
    "rating": 0, //op 5
    "pizzaId": "string"
}
```

De restaurant service maakt gebruik van MongoDB, de cluster staat ook online.



ePizza
Version 4.2.11

CLUSTER TIER
M0 Sandbox (General)

REGION
Azure / Netherlands (westeurope)

TYPE
Replica Set - 3 nodes

LINKED REALM APP
None Linked

Operations R: 0.01 W: 0.02 — 0.04/s — 0 — Last 6 Hours

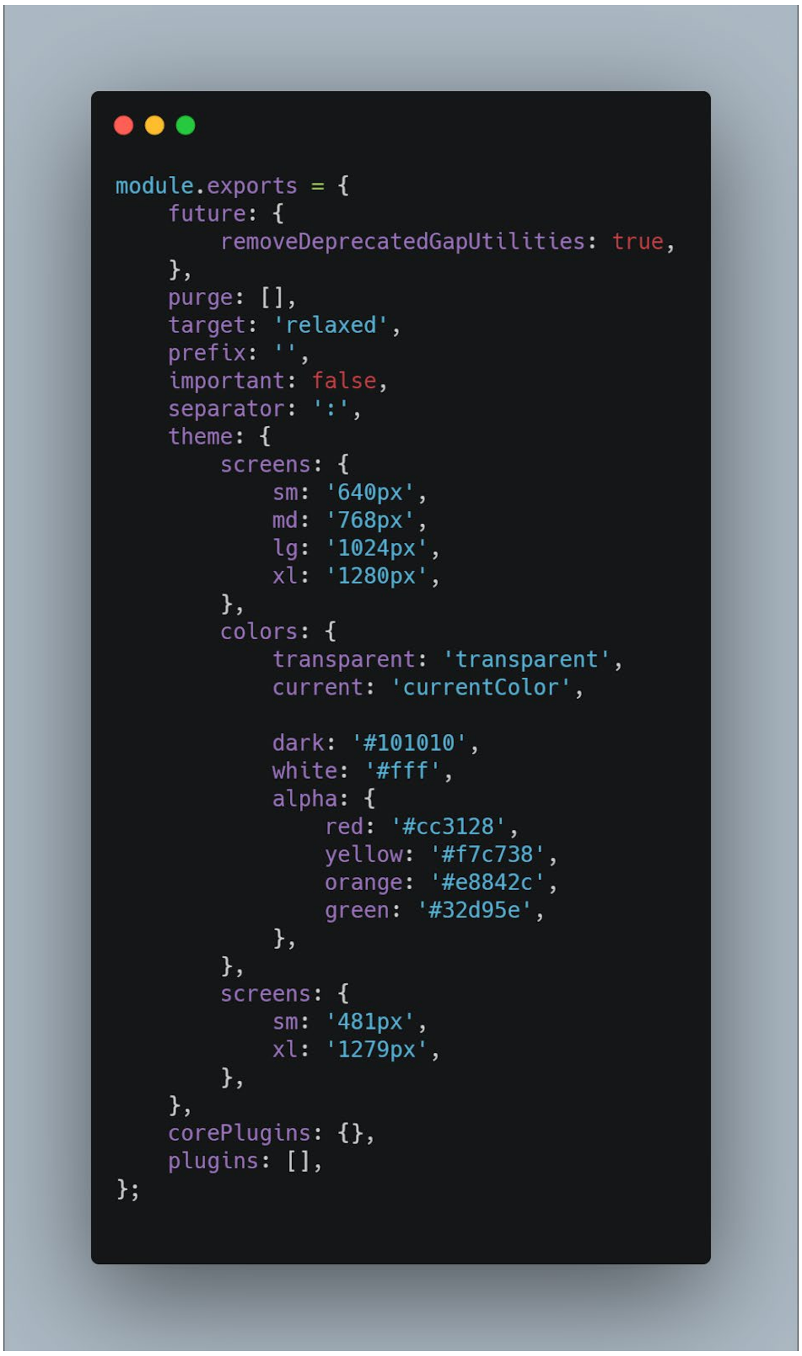
Logical Size 210.5 KB — 512.0 MB max — 0.0 B — Last 30 Days

Connections 15 — 500 max — 0 — Last 6 Hours

Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

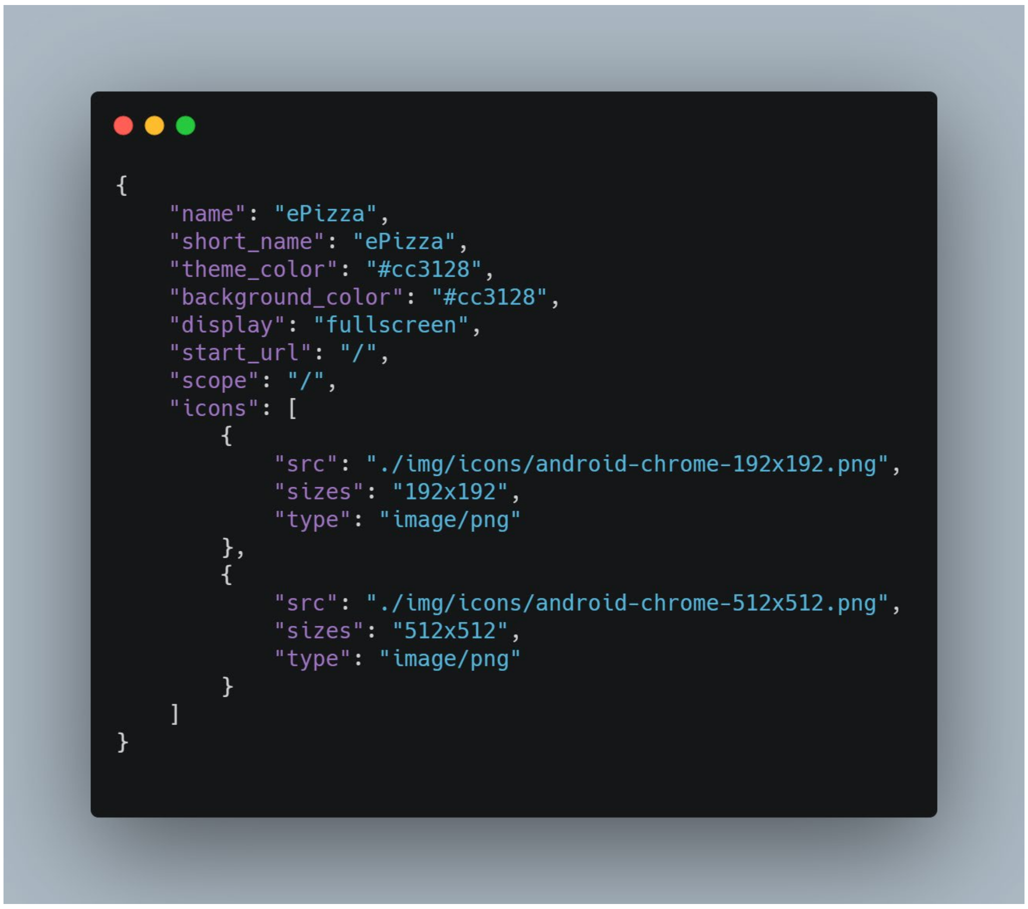
System Status: All Good
©2020 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Styling met Tailwind CSS



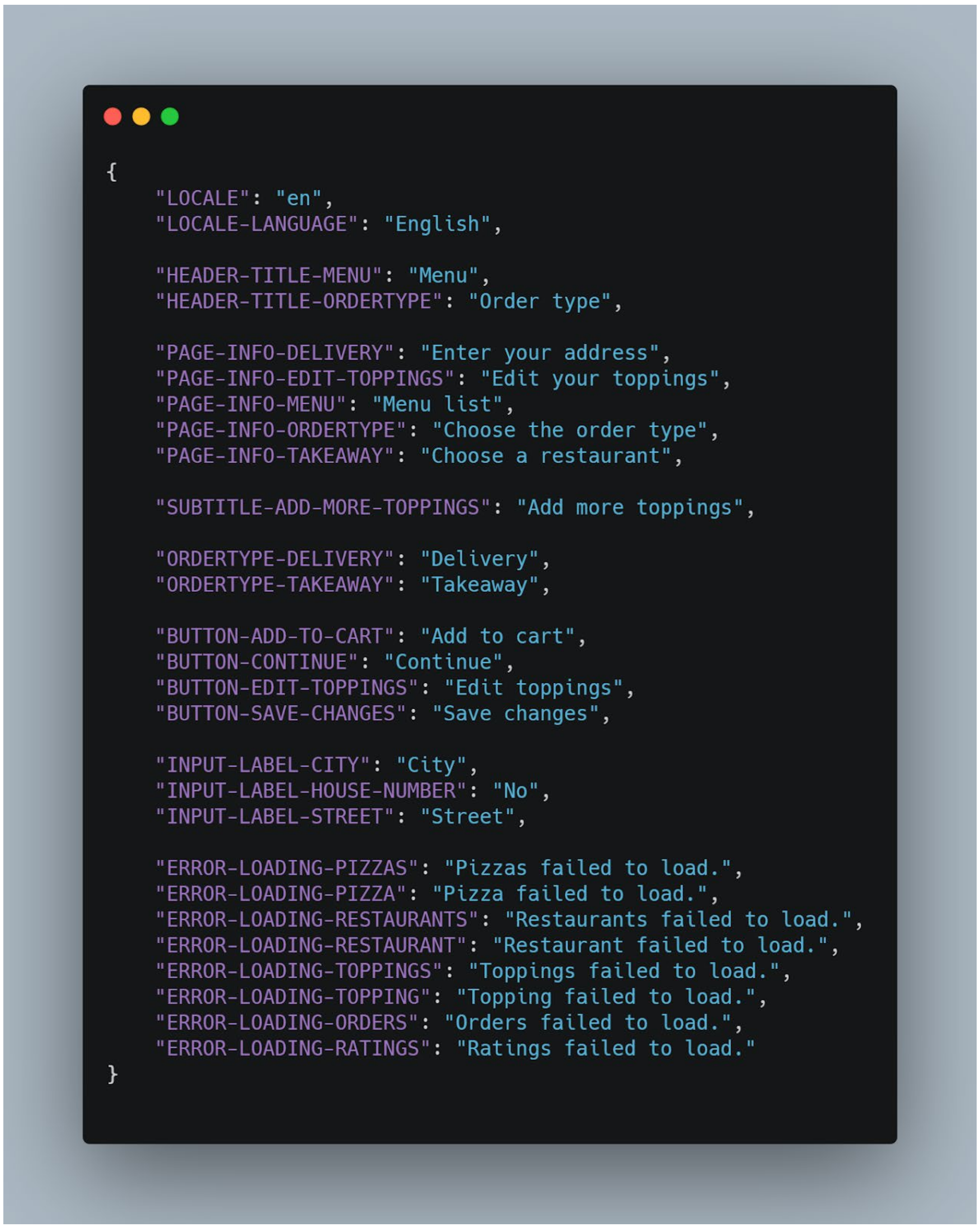
```
module.exports = {
    future: {
        removeDeprecatedGapUtilities: true,
    },
    purge: [],
    target: 'relaxed',
    prefix: '',
    important: false,
    separator: ':',
    theme: {
        screens: {
            sm: '640px',
            md: '768px',
            lg: '1024px',
            xl: '1280px',
        },
        colors: {
            transparent: 'transparent',
            current: 'currentColor',

            dark: '#101010',
            white: '#fff',
            alpha: {
                red: '#cc3128',
                yellow: '#f7c738',
                orange: '#e8842c',
                green: '#32d95e',
            },
        },
        screens: {
            sm: '481px',
            xl: '1279px',
        },
    },
    corePlugins: {},
    plugins: [],
};
```

PWA



```json
{
    "name": "ePizza",
    "short_name": "ePizza",
    "theme_color": "#cc3128",
    "background_color": "#cc3128",
    "display": "fullscreen",
    "start_url": "/",
    "scope": "/",
    "icons": [
        {
            "src": "./img/icons/android-chrome-192x192.png",
            "sizes": "192x192",
            "type": "image/png"
        },
        {
            "src": "./img/icons/android-chrome-512x512.png",
            "sizes": "512x512",
            "type": "image/png"
        }
    ]
}
```

Multi-language met i18n

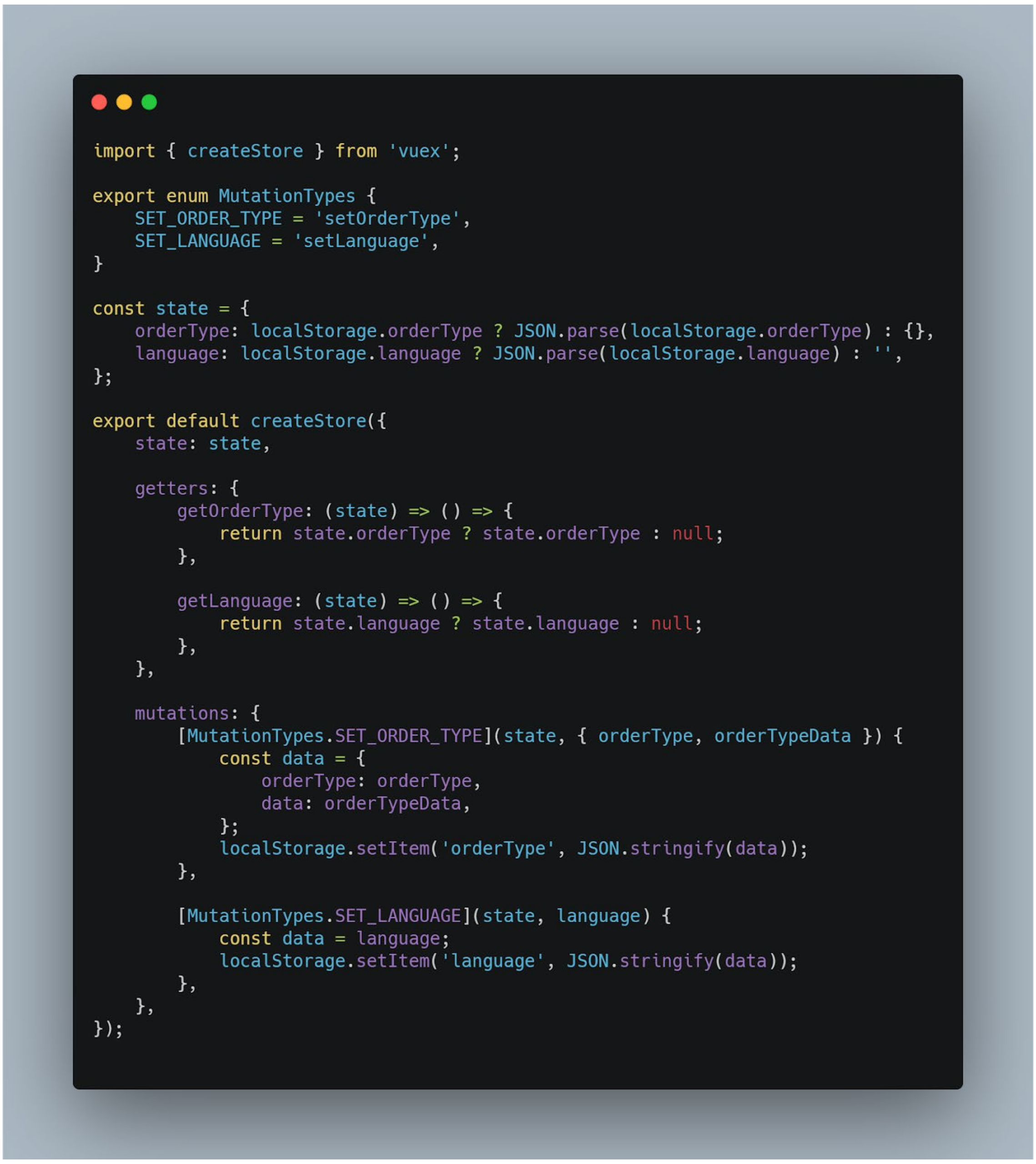


```json
{
    "LOCALE": "en",
    "LOCALE-LANGUAGE": "English",

    "HEADER-TITLE-MENU": "Menu",
    "HEADER-TITLE-ORDERTYPE": "Order type",

    "PAGE-INFO-DELIVERY": "Enter your address",
    "PAGE-INFO-EDIT-TOPPINGS": "Edit your toppings",
    "PAGE-INFO-MENU": "Menu list",
    "PAGE-INFO-ORDERTYPE": "Choose the order type",
    "PAGE-INFO-TAKEAWAY": "Choose a restaurant",

    "SUBTITLE-ADD-MORE-TOPPINGS": "Add more toppings",

    "ORDERTYPE-DELIVERY": "Delivery",
    "ORDERTYPE-TAKEAWAY": "Takeaway",

    "BUTTON-ADD-TO-CART": "Add to cart",
    "BUTTON-CONTINUE": "Continue",
    "BUTTON-EDIT-TOPPINGS": "Edit toppings",
    "BUTTON-SAVE-CHANGES": "Save changes",

    "INPUT-LABEL-CITY": "City",
    "INPUT-LABEL-HOUSE-NUMBER": "No",
    "INPUT-LABEL-STREET": "Street",

    "ERROR-LOADING-PIZZAS": "Pizzas failed to load.",
    "ERROR-LOADING-PIZZA": "Pizza failed to load.",
    "ERROR-LOADING-RESTAURANTS": "Restaurants failed to load.",
    "ERROR-LOADING-RESTAURANT": "Restaurant failed to load.",
    "ERROR-LOADING-TOPPINGS": "Toppings failed to load.",
    "ERROR-LOADING-TOPPING": "Topping failed to load.",
    "ERROR-LOADING-ORDERS": "Orders failed to load.",
    "ERROR-LOADING-RATINGS": "Ratings failed to load."
}
```

LocalStorage

Data in localStorage

- Order type (delivery/takeaway)
- Language

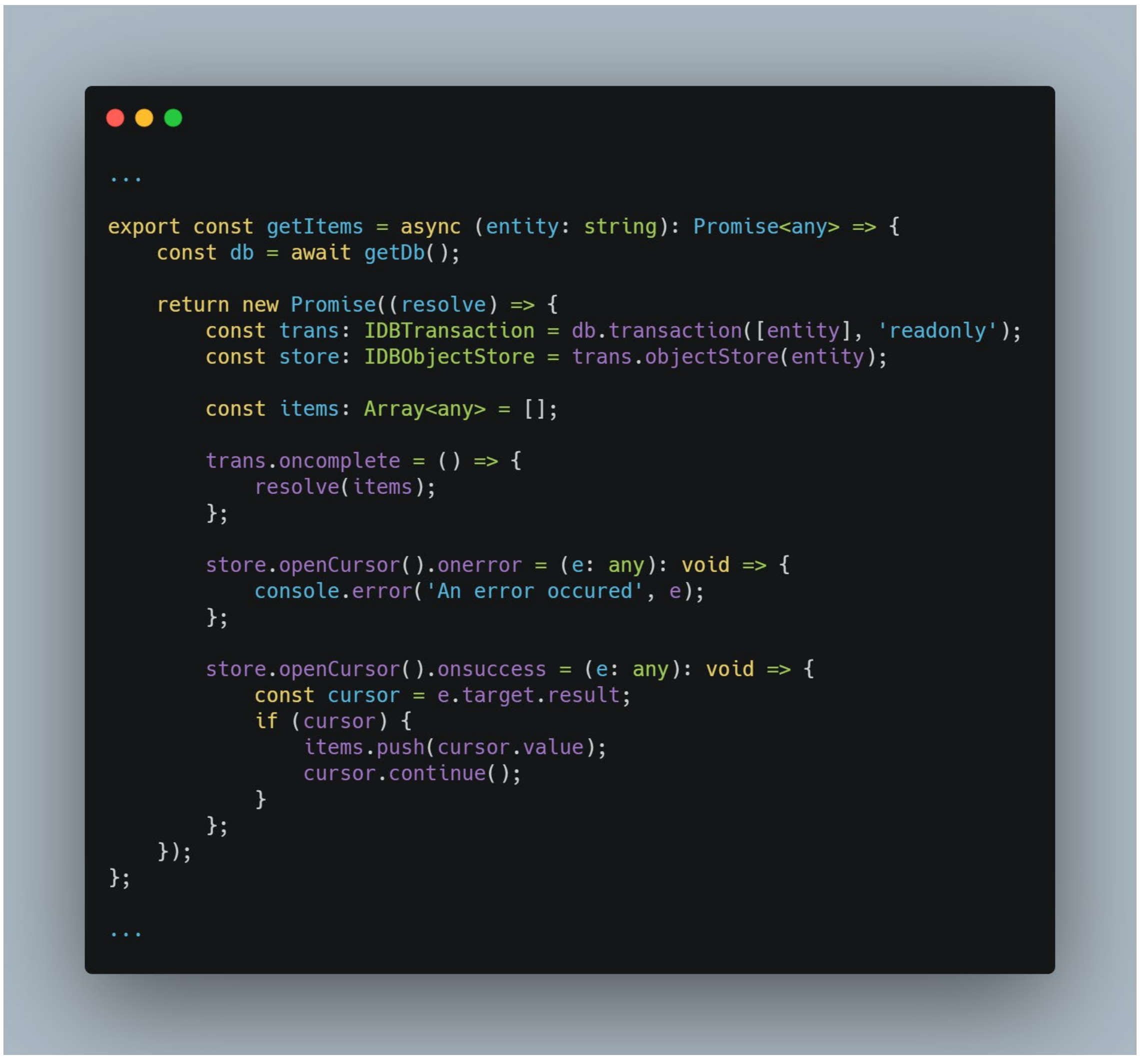


```
import { createStore } from 'vuex';

export enum MutationTypes {
    SET_ORDER_TYPE = 'setOrderType',
    SET_LANGUAGE = 'setLanguage',
}

const state = {
    orderType: localStorage.orderType ? JSON.parse(localStorage.orderType) : {},
    language: localStorage.language ? JSON.parse(localStorage.language) : '',
};

export default createStore({
    state: state,

    getters: {
        getOrderType: (state) => () => {
            return state.orderType ? state.orderType : null;
        },

        getLanguage: (state) => () => {
            return state.language ? state.language : null;
        },
    },

    mutations: {
        [MutationTypes.SET_ORDER_TYPE](state, { orderType, orderTypeData }) {
            const data = {
                orderType: orderType,
                data: orderTypeData,
            };
            localStorage.setItem('orderType', JSON.stringify(data));
        },

        [MutationTypes.SET_LANGUAGE](state, language) {
            const data = language;
            localStorage.setItem('language', JSON.stringify(data));
        },
    },
});
```

IndexedDB

Data in Idb

- Cart items (pizzas)

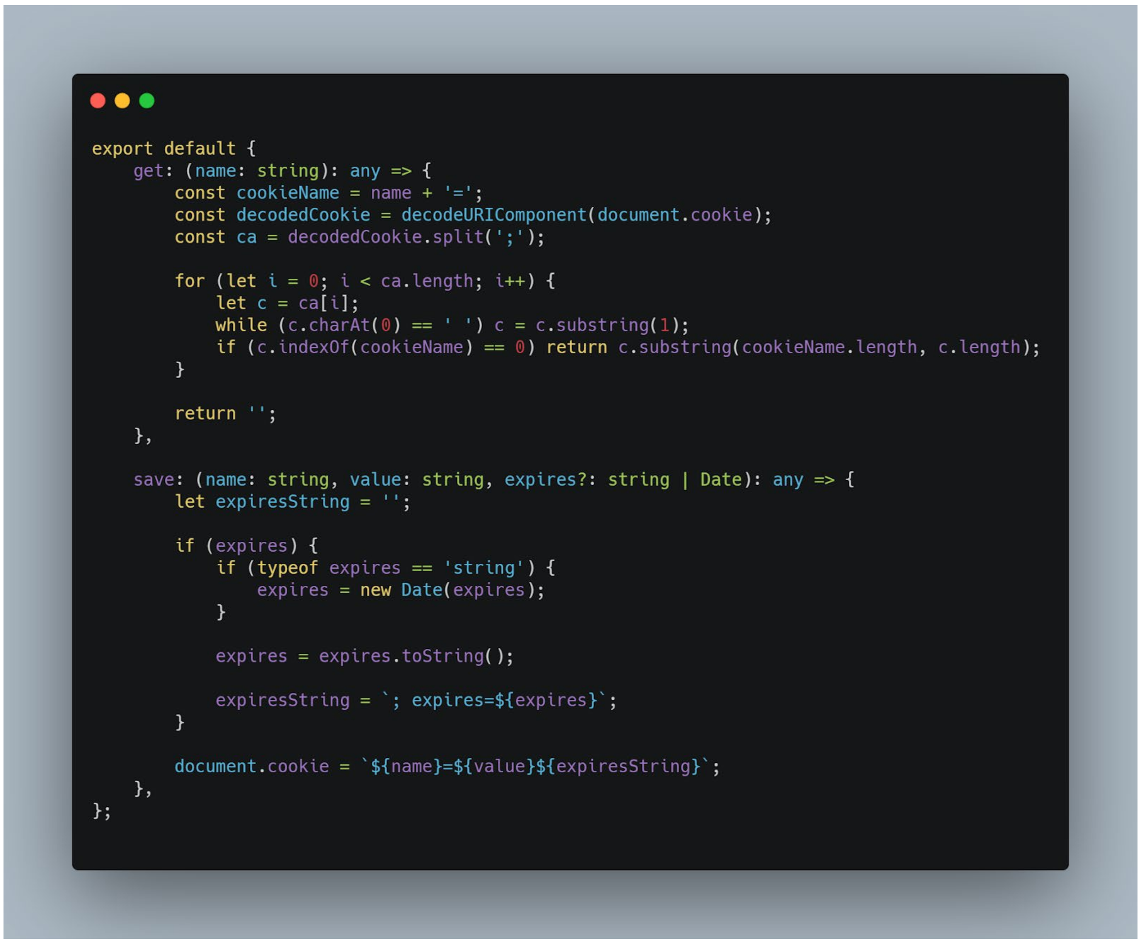


```
...

export const getItems = async (entity: string): Promise<any> => {
    const db = await getDb();

    return new Promise((resolve) => {
        const trans: IDBTransaction = db.transaction([entity], 'readonly');
        const store: IDBObjectStore = trans.objectStore(entity);

        const items: Array<any> = [];

        trans.oncomplete = () => {
            resolve(items);
        };

        store.openCursor().onerror = (e: any): void => {
            console.error('An error occured', e);
        };

        store.openCursor().onsuccess = (e: any): void => {
            const cursor = e.target.result;
            if (cursor) {
                items.push(cursor.value);
                cursor.continue();
            }
        };
    });
};

...
```

Cookies

Authenticatie

Het authenticatie token wordt opgeslagen in een cookie met een expiration

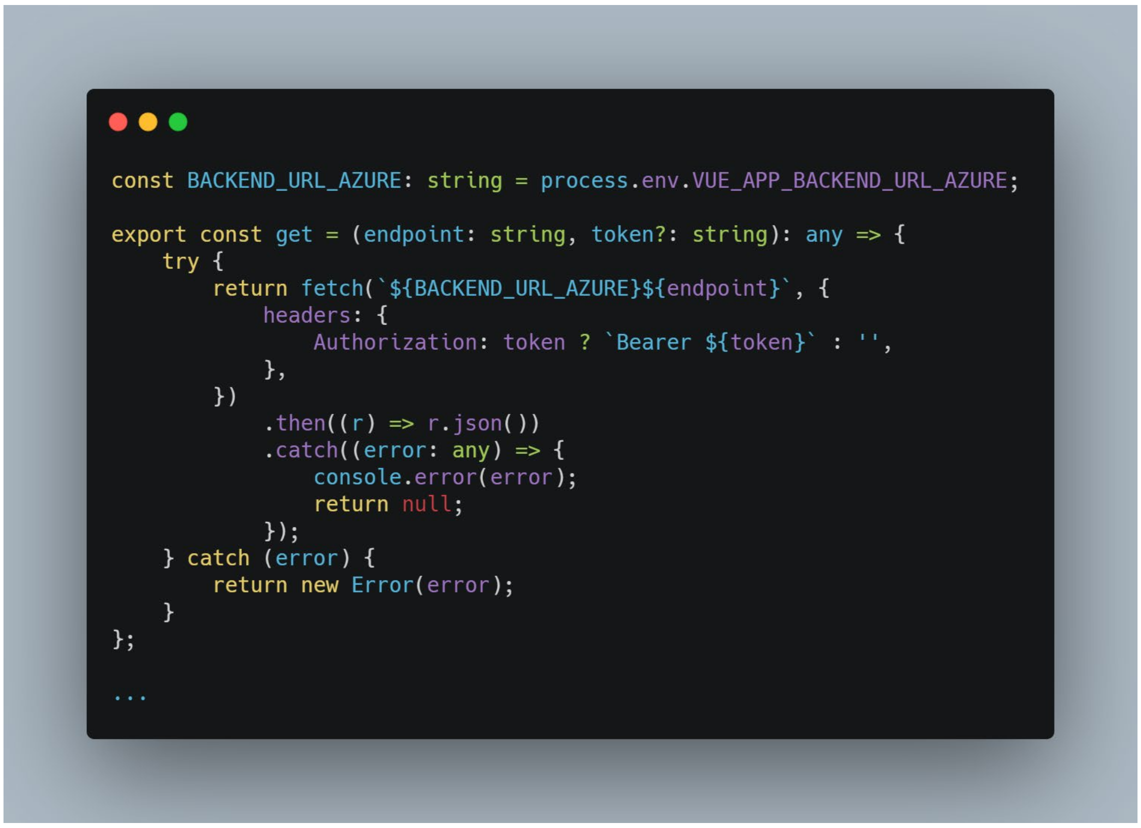


```
export default {
    get: (name: string): any => {
        const cookieName = name + '=';
        const decodedCookie = decodeURIComponent(document.cookie);
        const ca = decodedCookie.split(';');

        for (let i = 0; i < ca.length; i++) {
            let c = ca[i];
            while (c.charAt(0) == ' ') c = c.substring(1);
            if (c.indexOf(cookieName) == 0) return c.substring(cookieName.length, c.length);
        }

        return '';
    },

    save: (name: string, value: string, expires?: string | Date): any => {
        let expiresString = '';

        if (expires) {
            if (typeof expires == 'string') {
                expires = new Date(expires);
            }

            expires = expires.toString();

            expiresString = `; expires=${expires}`;
        }

        document.cookie = `${name}=${value}${expiresString}`;
    },
};
```

API

API mogelijkheden

- get(endpoint, token?)
- post(endpoint, payload, token?)
- put(endpoint, payload, token?)
- deleteById(endpoint, id, token?)



```
const BACKEND_URL_AZURE: string = process.env.VUE_APP_BACKEND_URL_AZURE;

export const get = (endpoint: string, token?: string): any => {
    try {
        return fetch(`${BACKEND_URL_AZURE}${endpoint}`, {
            headers: {
                Authorization: token ? `Bearer ${token}` : '',
            },
        })
            .then((r) => r.json())
            .catch((error: any) => {
                console.error(error);
                return null;
            });
    } catch (error) {
        return new Error(error);
    }
};

...
```

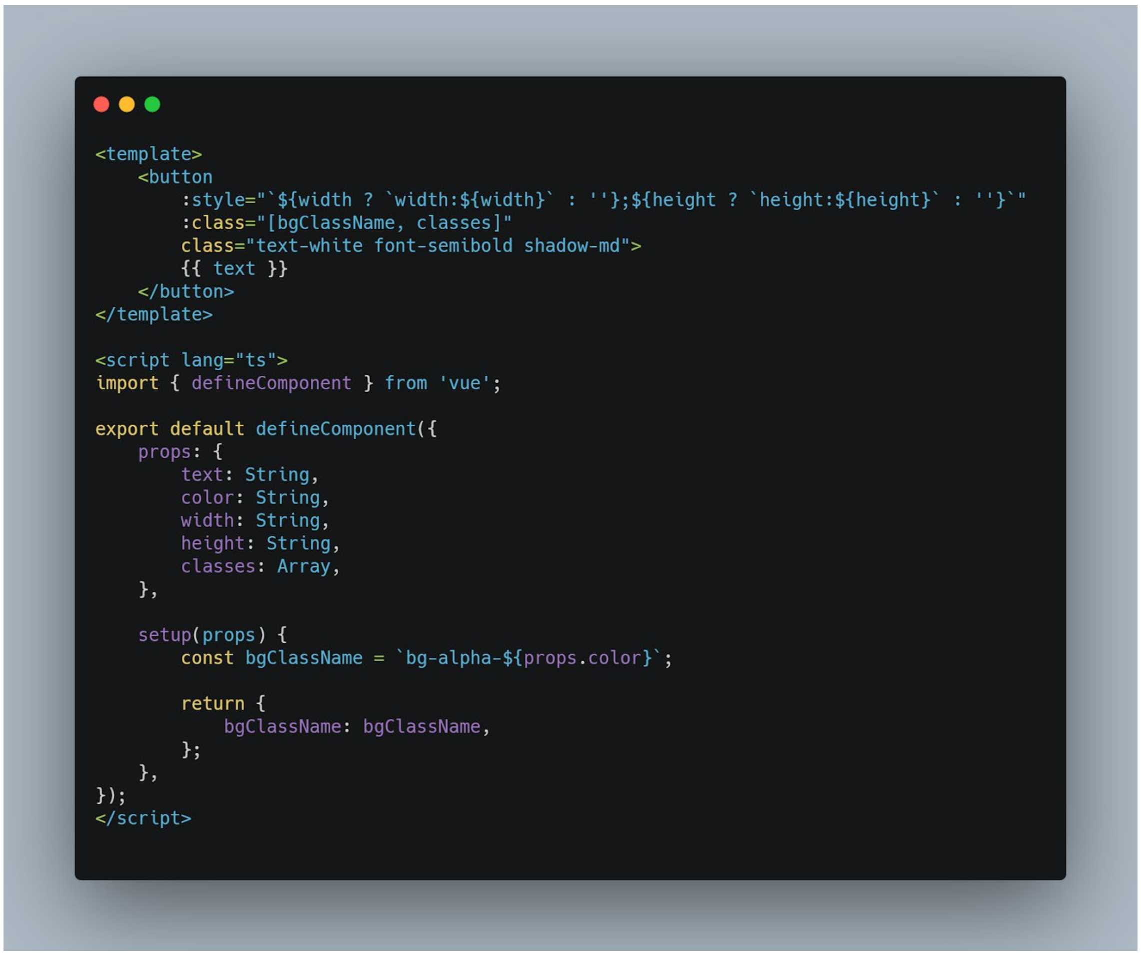
Custom reusable components

Customer components

- Rating

Shared Components

- Button
- DropdownList
- FooterComponent
- InputField
- LoadingIcon
- Navigationbar

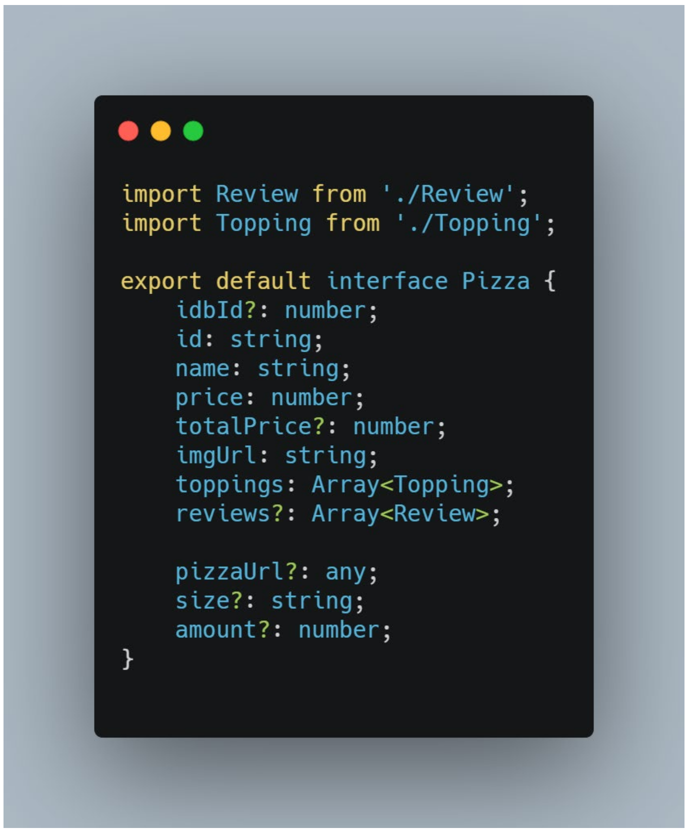


```
<template>
    <button
        :style="`${width ? `width:${width}` : ''};${height ? `height:${height}` : ''}`"
        :class="[bgClassName, classes]"
        class="text-white font-semibold shadow-md">
        {{ text }}
    </button>
</template>

<script lang="ts">
import { defineComponent } from 'vue';

export default defineComponent({
    props: {
        text: String,
        color: String,
        width: String,
        height: String,
        classes: Array,
    },

    setup(props) {
        const bgClassName = `bg-alpha-${props.color}`;

        return {
            bgClassName: bgClassName,
        };
    },
});
</script>
```

Models

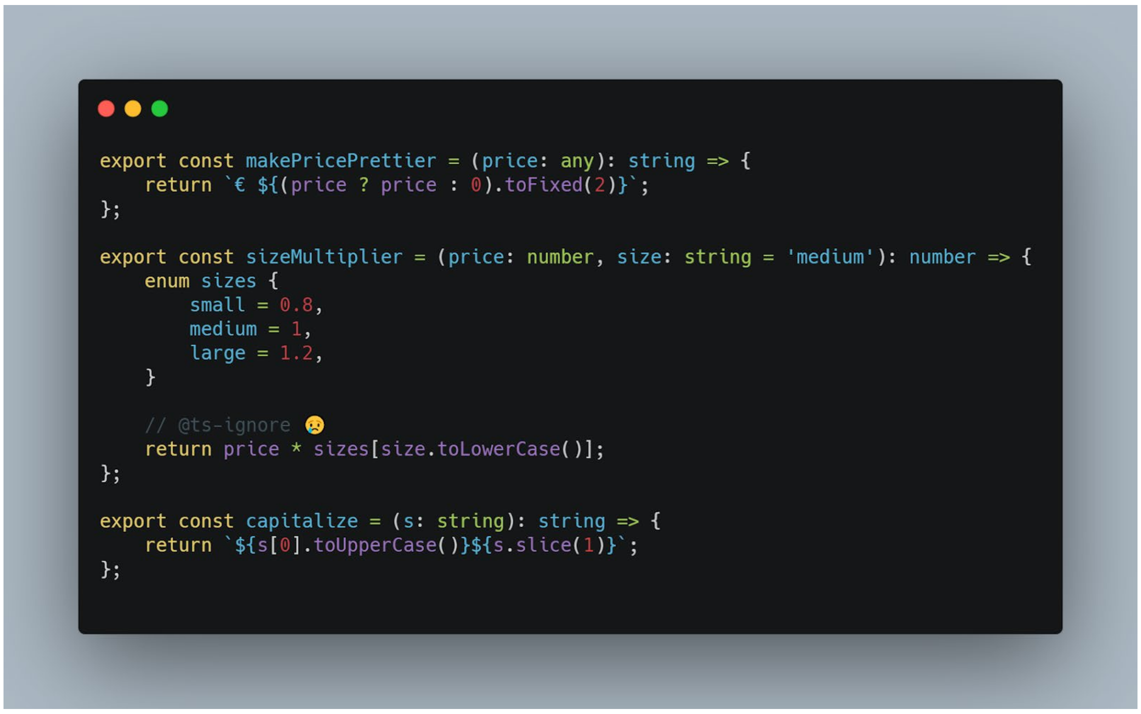
Models

- Customer
- Order
- Pizza
- Restaurant
- Review
- Topping



```
import Review from './Review';
import Topping from './Topping';

export default interface Pizza {
    idbId?: number;
    id: string;
    name: string;
    price: number;
    totalPrice?: number;
    imgUrl: string;
    toppings: Array<Topping>;
    reviews?: Array<Review>;

    pizzaUrl?: any;
    size?: string;
    amount?: number;
}
```

Reusable code



```
export const makePricePrettier = (price: any): string => {
    return `€ ${(price ? price : 0).toFixed(2)}`;
};

export const sizeMultiplier = (price: number, size: string = 'medium'): number => {
    enum sizes {
        small = 0.8,
        medium = 1,
        large = 1.2,
    }

    // @ts-ignore 😥
    return price * sizes[size.toLowerCase()];
};

export const capitalize = (s: string): string => {
    return `${s[0].toUpperCase()}${s.slice(1)}`;
};
```

howest
hogeschool

howest.be