

Tworzymy konfigurator ciastek w ciastkach :)

Adres tej dokumentacji to: [bit.ly/megak-konfigurator-ciastek](https://bit.ly/megak-konfigurator-ciastek)

## Dane wejściowe

Mamy dostępne dwie bazy ciastek:

- Żytnie - 5 zł
- Czekoladowe - 5 zł

Mamy bazę dodatków do ciastek:

- Czekolada - 5 zł
- Posypka - 3 zł
- Miód - 6 zł
- Żurawina - 4 zł
- Kokos - 10 zł

## Zasoby

Tutaj znajdziecie wszystkie assety:

<https://drive.google.com/file/d/1AxGnkOOIBKWc2sHBmHFBIZYmztN9za3Y/view?usp=sharing>

## Konfigurator

Chcemy stworzyć apkę, w której każdy może stworzyć swoje wymarzone ciastko. Może sobie wybrać jedną bazę, a także dowolną ilość dodatków.

Zauważ, że strona główna będzie renderowana przez Handlebars - a nie będzie statycznym index.html z plików publicznych.

Po wybraniu spodu i/lub dodaniu/usunięciu dodatku wyświetla się informacja z podziękowaniem za wybór spodu/dodatku i linkiem powrotu na stronę główną.

Na stronie głównej widać od razu nasze wymarzone ciastko. Wyświetla się także jego cena oraz to co na tę cenę wpłynęło. Z możliwością usunięcia danego składnika.

Podgląd ciastka i jego skład nie jest częścią szablonu, ale będzie potrzebny w innych miejscach - zastosuj partiale.

Wszystkie informacje na temat wybranego spodu czy dodatków przechowujemy w ciastkach (plikach cookie).

# Walidacja

- Kiedy dodajesz dodatek upewnij się, że taki dodatek istnieje.
- Kiedy dodajesz dodatek upewnij się, że taki dodatek nie został już dodany.
- Kiedy wybierasz bazę upewnij się, że taka baza istnieje.
- Kiedy usuwasz dodatek upewnij się, że taki dodatek został już dodany.

Jeżeli którykolwiek z warunków nie jest spełniony to wyświetlaj osobną podstronę błędu, która pokazuje czerwoną ramkę i opis problemu, a także link do strony głównej.

## Koszyk

Kolejnym modulem jest koszyk. Na stronie głównej powinien się pojawić przycisk z przejściem do koszyka.

Składa się on z tego samego widoku wszystkich elementów pokazujący ceny. Tylko bez możliwości usunięcia danego składnika. Żeby się nie powtarzać użyj tego samego partiala co na stronie głównej. W sprytny sposób przekaż, żeby nie wyświetlać możliwości usunięcia składnika.

PS. Możesz przekazać dodatkowe informacje do partiala w ten sposób:

```
{{> mypartial foo='bar' }}
```

Koszyk ma przycisk "Zamów", który na końcu wyświetla cenę ostateczną i podziękowanie za zakup.

Dodatkowo czyści to wszystkie ustawienia.

Na końcu podaje link do strony głównej.

## Dla ultra, bardzo wyjątkowo ambitnych (i z dużą ilością wolnego czasu) - nie na teraz

Zrobić wersję dynamiczną, opartą na fetchu.

Rozkminki

- **Konfiguracja Expressa**
  - JSON nie jest zbędny - póki nie używamy fetcha. Ale nie przeszkadza.
  - Pliki statyczne się przydadzą (opisane niżej).
  - cookie-parser niezbędny.
  - urlencoded (tradycyjne formularze) są zbędne.
  - hbs - potrzebne.
- **Ścieżki statyczne**
  - Usuwanie index.html.
  - Może się przydać plik CSS. A nawet na pewno.
  - Grafiki będą potrzebne.
- **Routers / moduły**
  - configurator.
  - home.
  - order.
- **Widoki / ścieżki**
  - Strona główna.
  - Podziękowanie za dodanie dodatku.
  - Podziękowanie za usunięcie dodatku.
  - Podziękowanie za wybór spodu.
  - Strona błędu.
  - Podsumowanie koszyka.
  - Podziękowanie za zakupy.
- **Partiale**
  - Lista wybranych dodatków i ich cen.
  - Podgląd ciastka.
  - Powrót do strony głównej.
- **Przechowanie informacji o aktualnym zamówieniu**
  - Ciastko ze spodem: string.
  - Ciastko z dodatkami: tablica stringów. JSON.
- **Przechowanie bazy danych o spodach i dodatkach**
  - W pliku .js będą dwa obiekty. Jeden ze spodami, a drugi z dodatkami.
  - W obiektach klucze to będą nazwy spodu/dodatku, a wartości to będą ceny.
- **Sposób wyświetlania ciastka**
  - div ze stałą szerokością i wysokością ze wstawionymi wieloma obrazkami nałożonymi na siebie przez background.

PS. Gotowa wersja **zajrzyj dopiero po wszystkich odcinkach z robienia kodu!!**

<https://replit.com/@Ami777/MegaK-Professional-Cookie-Maker#index.js>

Wersja OOP: <https://replit.com/@Ami777/MegaK-Professional-Cookie-Maker-OOP#index.js>

Schemat do OOP:

