

# Teas API - Kliens

Rostagni Csaba

## Előkészületek

1. Készítsen egy fájlt `teas.html` néven ami egy szabványos HTML5 oldal szerkezetét tartalmazza. A nyelvet állítsa magyarra, míg a karakterkódolást `utf-8`-ra.
2. A fájlhoz linkelje hozzá a Bootstrap CSS fájlt.
3. Hozzon létre egy fájlt `teas.css` néven a `css` mappában, majd azt is linkelje a HTML oldalhoz.
4. A szkriptnek hozzon létre egy fájlt `teas.js` néven a `js` mappában. Ne felejtse szigorú szabványos módot megkövetelni (`"use strict"`). Az így elkészült JS fájl is kösse össze a HTML oldallal.
5. Egy fix szélességű tárolóelemben Címsor 1-es kiemeléssel jelenítse meg az oldal címét: “Teák”.
6. Ugyanez a szöveg jelenjen meg a böngészőfülön is.
7. A cím után hozzon létre egy keretet (`div`), aminek az azonosítója legyen `teas`.

## Táblázat

### Fejléc

8. A teák táblázatának megjelenítéséhez hozzon létre egy függvényt `generateTable` néven, aminek egyetlen paramétere van, a teák tömbje (`teas`).
  - Hozzon létre egy új táblázat elemet egy `table` nevű változóban.
  - Állítsa be a táblázatra, hogy kapja meg a Bootstrap csíkozott formázását.
  - Hozza létre a táblázat fejlécét egy `thead` nevű változóban.
  - A táblázat fejléce összesen egy sorból áll. Hozzon létre egy `tr` nevű változóban.
  - Összesen 5 fejléc elemere lesz szükség, rendre: “*Megnevezés*”, “*Gyártó*”, “*Fajta*”, “*Kiszerelés*”, “*Mennyiség*” és “*Ár*” feliratokkal. Készítsen egy tömböt `titles` néven, ami eltárolja ezeket.
    - Mivel lényegiben többször kell pontosan ugyanazt csinálni, így ezt egy ciklussal célszerű megoldani.
    - Az egyetlen ami változik, az maga a felirat.
  - Készítsen egy ciklust, ami bejárja az imént elkészített tömböt.
  - A ciklusban hozzon létre egy táblázat fejléc elemet egy `th` nevű változóba.
  - A fejléc elem szövegének adja meg az aktuális elemét a tömbnek
  - Ahhoz, hogy majd a táblázatban megjelenhessenek hozzá kell adni a táblázat sorához minden egyes fejléc elemet.
  - Miután elkészült a fejléc sora, a táblázat fejlécéhez hozzá kell adni a sort.
  - A fejléc elkészültével adja hozzá az egészet a táblázathoz.

- Bár még nem tartalmaz adatot a táblázat, de a tesztelhetőség miatt a függvény adja vissza az elkészült táblázatot.

## Táblázat megjelenítése

9. Készítsen egy függvényt `displayTable` néven, aminek a feladata annyi, hogy hívja meg a szükséges generáló függvényt, majd a kapott eredményt jelenítse meg a `teas` azonosítójú elemében.
10. Hogy tesztelje az imént elkészített függvényt, hívja meg.

## Tartalmi rész

11. A táblázat sorainak összeállításához hozzon létre egy függvényt `generateRow` néven, aminek a bemenete (`tea`) egy olyan objektum, ami egy adott tea adatait tartalmazza.
  - Hozzon létre egy sort, egy `tr` nevű változóban.
    - Az azonos változónév **nem fog összeakadni** a másik függvényben létrehozottal, mert **másik scope**-ban vannak.
  - A fejléchez hasonlóan hozzon létre egy tömböt `contents` néven, ami a megjelenítendő adatokat tartalmazza a minta szerint.
  - A fejléchez hasonlóan járjon el a cellák létrehozásakor is. Hozzon létre egy új táblázat vella elemet, majd miután beállította a szöveges tartalmát adja hozzá a táblázat sorához.
    - Jelen esetben egyetlen sorunk van, így a `tr` egyértelmű
  - Végezetül a függvény térjen vissz az elkészült sorral.
12. Bővítse ki a `generateTable` függvényt, hogy ne csak a fejléctet, hanem a táblázatot tartalmi részét is állítsa össze
  - Hozza létre a táblázat törzsét egy `tbody` nevű változóban.
  - Járja be a paraméterül kapott teákat, majd generáltassa le az adott sort és adja is hozzá a táblázat törzséhez.

## Táblázat tesztelése

13. A táblázatban az adatok csak akkor jelennek meg, ha a `generateTable` függvény megkapja paraméterként a teák tömbjét `teas` néven. A teszteléshez hozzon létre egy tömböt a `displayTable` függvényen belül az alábbi adatokkal:

name	range	format	qty	unit	price
English Breakfast	black	tea bag	25	db	1469
Earl Grey	black	tea bag	25	db	1469

A brand mind a két tea esetében az alábbi legyen:

```
{
  "id": 3,
  "name": "Twinings",
  "country": "Egyesült Királyság"
}
```

## API - Teák lekérése

14. Az API eléréséhez szükség lesz az elérhetőségére, ezt tárolja el `baseUrl` néven a fájl elején.
15. A teszteléshez megfelelő a tömb, de a valós működéshez az API-tól kell a kiindulóponttól számítva a `/teas` csompóponttól betölteni az adatokat. Módosítsa a `displayTable` függvényt úgy, hogy indítson egy kérést a Fetch API segítségével a backendnek.
  - Amennyiben a szerver által küldött adat sikeresen megérkezik, úgy a válasz törzsrészből kiolvasott adatokat JSON-ként értelmezze. Míg hiba esetén “*Hálózati hiba*” üzenetet adjon.
  - Amint az adatok értelmezése sikeresen megtörtént a kapott `teas` tömbbel hívja meg a `generateTable` függvényt, és az általa generált táblázatot jeleztse meg a `teas` azonosítójú elemben. Innentől a korábbi tesztкод legyen kikommentelve.
  - Hiba esetén írja ki a konzolra a hibát, de ne egyszerű üzenetként, hanem hibaként, majd a felhasználók számára egy felugróablakot is jelenítsen meg: “*A teák betöltése sikertelen*” üzenettel.

A teljes `displayTable` függvény:

Most már a táblázat helyett az API-ból kapott adatoknak kell megjelennie.

## Űrlap létrehozása

16. A teákat megjelenítő keretet (`div`) követően hozzon létre egy űrlapot.
17. A tea nevének megadásához hozzon létre egy szöveges beviteli mezőt `name` névvel és azonosítóval, továbbá a hozzá tartozó címkét. Minden beviteli mezőt és a hozzá tartozó címkét egy-egy külön keretben (`div`) helyezze el!
  - Állítsa be az űrlapelem alapértelmezett értékét “*Tea*”-ra.
  - HTML 5-ös validációval határozza meg, hogy a tea neve legyen legalább 1 karakter, de legfeljebb 90 karakter hosszú.
18. Hozza létre a legördülőlistát a gyártóhoz `brand_id` névvel és azonosítóval, továbbá a hozzá tartozó címkét. Az értékeit az alábbi táblázat alapján határozza meg, a megadott sorrendben!

Kulcs	Érték	Alapértelmezett
1	Lipton	
2	Tetley	
3	Twinings	igen

19. Hozza létre a legördülőlistát a típushoz `range` névvel és azonosítóval, továbbá a hozzá tartozó címkét. Az értékeit az alábbi táblázat alapján határozza meg, a megadott sorrendben!

Kulcs	Érték	Alapértelmezett
black	Fekete tea	
fruit	Gyümölcs tea	igen
green	Zöld tea	

20. Hozza létre a legördülőlistát a kiszínezéshez `format` névvel és azonosítóval, továbbá a hozzá tartozó címkét. Az értékeit az alábbi táblázat alapján határozza meg, a megadott sorrendben!

Kulcs	Érték	Alapértelmezett
tea bag	Filteres	
can	Fémdobozos	
bottle	Palack	igen

21. A mennyiség megadásához hozzon létre egy szám eltárolására alkalmas beviteli mezőt **qty** névvel és azonosítóval, továbbá a hozzá tartozó címkét.

- Állítsa be az űrlapelem alapértelmezett értékét “1”-re.
- HTML 5-ös validációval határozza meg, hogy a mennyiség legalább 1, de legfeljebb 3000 lehessen.

22. Hozza létre a legördülőlistát az egységhez **unit** névvel és azonosítóval, továbbá a hozzá tartozó címkét. Az értékeit az alábbi táblázat alapján határozza meg, a megadott sorrendben!

Kulcs	Érték	Alapértelmezett
db	darab	
l	liter	
ml	mililiter	igen

23. Az ár megadásához hozzon létre egy szám eltárolására alkalmas beviteli mezőt **price** névvel és azonosítóval, továbbá a hozzá tartozó címkét.

- Állítsa be az űrlapelem alapértelmezett értékét “380”-ra.
- HTML 5-ös validációval határozza meg, hogy a mennyiség legalább 1, de legfeljebb 5000 lehessen.

24. Végezetül hozzon létre egy elküldésre alkalmas gombot **ment** azonosítóval, és “Ment” felirattal.

25. A CSS fájlban formázza meg az űrlapot, hogy elkülönüljön a többiektől. A **teas** azonosítójú elem után közvetlen következő **form** elemre állítsa be az alábbi formázásokat:

- Kapjon egy képpont folytonos fekete szegélyt
- Kapjon 70%-ban áttetsző fekete, vetett árnyékot az X és az Y tengelyen is 3-3 képpontal elvolva, míg az elhomályosítás mértéke legyen 10 képpont.
- Kapjon az alapértelmezett betűméret felének megfelelő méretű margót
- Kapjon az alapértelmezett betűméret felének megfelelő méretű belső margót

## Új tea létrehozása

26. Hozzon létre egy eseményt feldolgozó függvényt **sendNewTea** néven, aminek egyetlen bemeneti paramétere az esemény objektum (**evt**) legyen.

- Állítsa be, hogy a gomb ne hajtsa végre az alapértelmezett viselkedését.

- Kerresse ki egy-egy változóba az űrlapelemeket. A nevük legyen a mező nevet egy input szövégződésse
- Hozzon létre egy új objektumot `newTea` néven, ami az alábbi kulcsokhoz tartozó megfelelő értékeket
- Kulcsok: `name`, `brand\_id`, `range`, `format`, `qty`, `unit`, `price`
- Az API-tól kell a kiindulóponttól számítva a `/teas` csompópontnak elküldeni az adatokat.
- A kérés metódusa legyen `POST`.
- Állítsa be a kérés fejlécében, hogy a küldött és a fogadni kívánt adat típusa JSON.
- A kérés törse legyen a `newTea` objektum JSON string megfelelője.

- A válasz megérkezésekor ellenőrizze, hogy az sikeres kóddal tért e vissza. Amennyiben nem, úgy az e
- Amennyiben minden rendben volt olvassa ki a válasz törzséből az adatokat és adjva vissza JSON objektu
- Amennyiben minden sikeres volt úgy a `generateRow` segítségével hozzon létre egy új sort a táblázat
- Amennyiben bármilyen hiba keletkezett a feldolgozás során, azt jelenítse meg hibaként a konzolon.
- Keresse meg az űrlapot és tárolja el egy `form` nevű változóban.
- Az űrlap elküldéséhez adjon hozzá egy eseményt, ami a `sendNewTea` függvényt hajtja végre.

## Új gomb

Az űrlap legyen alapértelmezetten elrejtve, az új gombra kattintva jelenjen meg a teák alatt és görgessen oda az oldal.

## Törlés

Egészítse ki a táblázatot egy “Admin” oszloppal. Minden egyes tea sorában jelenjen meg egy “Törlés” gomb. A gomb data attribútumában tárolja el a tea azonosítóját. A gombra kattintás eseménye hajtson végre egy közös függvényt, ami API híváson keresztül, DELETE metódussal törli a rekordot, majd az aktuális sort is eltávolítja a táblázatból.

## Módosítás

Bővítse az admin oszlopot egy módosítás gombbal. A gombra kattintva az adott tea adatait töltse be az űrlapba. Bővítse ki az űrlapot egy “Módosít” gombbal, amire kattintva elküldi PU/PATCH metódussal az adatokat, beleértve az ID-t is.