

# Car registration

Promise, ESM

Rostagni Csaba

1. Klónozza le az alapsomagot `vezeteknev-keresztnev-car-registration` néven.
2. Másolja be a forrásban megadott `index.html` fájlt a projekt gyökerébe.
3. Másolja be a forrásban megadott `cars.js` és `peresons.js` fájlokat az `src/data` mappába.

## `cars.js`

4. A `cars.js` fájlban hozzon létre egy függvényt `findCarByPlate` néven, aminek egy bemeneti paramétere van, maga a rendszám.
5. Hozzon benne létre egy Promise-t, ami
  - a. Ellenőrizze, hogy kapott-e rendszámot, ha nem, akkor utasítsa el egy új `Error` objektummal, aminek a tartalma: *"plate is required* üzenettel. Szakítsa meg a függvény futását.
  - b. A `cars` tömbben keresse meg a rendszám alapján az autót.
  - c. Ellenőrizze, hogy van-e találat. Ha nincs, akkor utasítsa el egy új `Error` objektummal, *"car not found: RENDSZÁM* üzenettel. Szakítsa meg a függvény futását.
  - d. Végül, ha minden rendben volt, akkor teljesítse a kérést a megtalált `car` objektummal.
  - e. A modul a fenti függvényt elnevezetten tegye elérhetővé.

## `main.js`

6. Keresse meg az alábbi azonosítóval ellátott elemeket, és tárolja el egy változóban: `main`, `error`, `reset`, `search`.
7. A `main.js` fájlban bővítse a `createCarElement` függvényt, ami paraméterül kap egy objektumot.
8. Hozzon létre egy-egy új elemet a korábban elkészített `createStatCard` felhasználásával, az alábbi label - value formátumban megadott adatokkal ellátva:
  - `Year` - `year`
  - `Plate` - `plate`
  - `Owner` - `Click here to show owner info`
9. Az utoljára létrehozott elemhez adjon hozzá egy eseményfigyelőt, a kattintás eseményére:
  - a. Ezen a ponton csak Törölje ki saját magát, később majd tölts be a tulaj adatait és jelenítse meg.
    - Ehhez olvassa ki és tárolja el egy változóban, hogy melyik melyik elemre kattintottak.
10. A cikkhez adja hozzá a címet és a definíciós listát is, majd adja vissza az így létrejött elemet.
11. A `search` azonosítójú elem inputjának változásakor
  - a. Űrítse ki `main` azonosítójú elemet.
  - b. Hívja meg a `findCarByPlate` függvényt, amennyiben sikeres, úgy
    - Az `error` azonosítójú elemhez adja hozzá a `hidden` osztályt
    - Hozzon létre a `createCarElement` felhasználásával egy új autó megjelenítésére szolgáló elemet és helyezze el a `main` azonosítójú elembe.
  - c. Hiba esetén az `error` azonosítójú elembe található bekezdés szövege legyen a paraméterként kapott objektum `message` kulcsú értéke
    - Vegye le a `hidden` osztályt az `error` azonosítójú elemről.

## persons.js

12. A `persons.js` fájlban hozzon létre egy függvényt `findPersonById` néven, aminek egy bemeneti paramétere van, egy személy azonosítója-rendszer.
13. Hozzon benne létre egy Promise-t, ami
  - a. Ellenőrizze, hogy kapott-e azonosítót, ha nem, akkor utasítsa el egy új `Error` objektummal, aminek a tartalma: *“person\_id is required* üzenettel. Szakítsa meg a függvény futását.
  - b. Olvassa ki az azonosító számértékét, majd a `persons` tömbben keresse meg az azonosító alapján.
  - c. Ellenőrizze, hogy van-e találat. Ha nincs, akkor utasítsa el egy új `Error` objektummal, *“person not found: AZONOSÍTÓ* üzenettel. Szakítsa meg a függvény futását.
  - d. Végül, ha minden rendben volt, akkor teljesítse a kérést a megtalált `person` objektummal.
  - e. A modul a fenti függvényt elnevezetten tegye elérhetővé.

## Személy betöltése

14. A *“Click here to show owner info”* feliratú gombra kattintva
  - a. indítson egy kérést, hogy az adatokat betöltse `findPersonById` alkalmazásával. Az azonosító a `car` objektum `owner_id` kulcsában található
15. Amint visszaérkezett a válasz, úgy a `createStatCard` felhasználásával hozza létre a nevet és e-mail címet tartalmazó elemeket.
  - Ügyeljen rá, hogy a törlés csak akkor menjen végbe, ha sikeresen megjött a tulajdonos adata
  - Hiba esetén ugyanúgy járjon el, mint a másik Promise hívásnál