



Üzleti szoftverfejlesztés a gyakorlatban

VII. Gyakorlat - Play Framework

Gyakorlatvezető: János Zsolt

zsolt@worldofbooks.com

World of Buzz Ltd.



Formok

- A HTML oldalakon formokkal tudunk user input elkérni és azt a szerverre küldeni
- Definálható vele:
 - A szerver oldali cím, ahová az adatot akarjuk küldeni
 - Az inputok - kvázi a beviteli mezők, amiket látni kívánunk
 - A http metódus (GET vagy POST)

Form elemek

- Szöveges inputok
- Hidden inputok
- Jelszavas inputok
- Checkboxok
- Radiobutton
- Select (dropdown)
- A HTML5-ben sok új elem is megjelent (szín, dátumválasztó) amiket korábban “kézzel” kellett megoldani

HTTP metódusok

- A HTTP egy kérdés-válasz alapú protokoll - a kliens kérdésére a szerver küld egy választ
- A HTTP metódus az a kérdés módját definiálja
- GET = adatkérés
- POST = adatküldés
- DELETE = adat törlése
- Jelzésértékű alapvetően, nem minden szerver tartatja be

HTTP metódusok

GET	POST
Az adat az URL-ben utazik: <code>http://server.com/getSomething?param1=value1</code>	Az adat a HTTP üzenet body részében utazik, tehát az URL-ben nem látszanak a paraméterek
Cacheelhető	Nem cachelik a szerverek, böngészők
Csak szöveges paramétereket lehet vele küldeni	Bináris adatot is küldhetünk vele (fájl feltöltés esetén pl)
Nem szabad benne érzékeny adatot küldeni	Ezt használják jelszók szerverre küldésére
Általában keresésekre, linkelhető tartalmakra, adatot nem változtató kérésekre használják	Általában olyan requestek esetén használják, ami adatot módosít vagy hoz létre a szerveren (regisztráció, üzenetküldés, stb)

Play framework és a validáció

- A szerverre küldött adatokat sosem elég csak a kliens oldalon (javascripttel például) validálni, a serveren is mindig ellenőrizni kell az adatokat
 - Így garantálható, hogy az üzleti logikának megfelelő állapotba kerül és marad a rendszer
- A Play-nek saját validációs megoldása van, ami megkönnyíti a hibák regisztrálását, figyelését és a template (.html) fájlbeli feldolgozását
- Minden controller osztályban elérhető a validation nevű változó
- Ezt a validation változót kell feltölteni a hibákkal és figyelni a tartalmát az események végrehajtása előtt

Feladat 1

- Készítsünk egy új képernyőt, ami a GET /createLibrary hívásra jön be
- A képernyőn egy formot készítünk, a következő inputokkal:
 - Könyvtár neve (szöveges)
 - Könyvtár irányítószáma (szöveges)
- A form a POST /createLibrary HTTP endpointra küldje az adatot
- A szerver oldalon validáljuk a következőket:
 - A könyvtár neve nem üres
 - A könyvtár irányítószáma 1000 és 10000 közötti szám
 - A könyvtár a megadott néven nem létezik már a rendszerben
- Ha a validáció sikertelen, akkor dobjuk vissza a formra a requestet és írjuk ki a validációs hibákat
- Ha a validáció sikeres, akkor mentjük le a Library entitást a kapott adatokkal
- A request legvégén irányítsunk át a GET /libraryBooks oldalra
- Egészítsük ki a GET /libraryBooks oldalt egy "Könyvtár létrehozása" linkkel, ami a GET /createLibrary oldalra irányít.

Feladat 2

- Alakítsuk át a GET /libraryBooks oldalt úgy, hogy az oldalon csak egy táblázat van, két oszloppal:
 - Könyvtár neve
 - Egy link: “Könyvtáradatok megjelenítése”
- A link mutasson egy GET /libraryDetails?libraryId=.. Oldalra
- Ezen az oldalon jelenjen meg az az adat az adott könyvtárról, ami eddig a GET /libraryBooks oldalon
- A GET /libraryDetails oldal alján legyen egy link, ami visszavisz a GET /libraryBooks oldalra

Linkek

- Play validáció:
<https://www.playframework.com/documentation/1.4.x/validation>
- http://www.w3schools.com/tags/ref_httpmethods.asp