



Struts – példaprogram

Adott egy egyszerű fórumot megvalósító alkalmazás, melyben az alábbiakra van lehetőség:

- új felhasználó regisztrálása
- regisztrált felhasználó bejelentkezése
- témakörök és hozzászólások böngészése (akkor is, ha a felhasználó nincs bejelentkezve)
- új témakör illetve hozzászólás hozzáadása (ha a felhasználó be van jelentkezve)

Bővítsük ki az alkalmazást a következő követelmények alapján...

- A `topics.jsp`-ben található táblázat fejléceit alakítsuk linkeké és az illető oszlopra hívjuk meg a megfelelő rendezést.
- A rendezés úgy működjön, hogy ha ugyanarra az oszlopra többször kattintunk, akkor változzon a rendezési sorrend (növekvő/csökkenő).

módosítások:

- `ViewTopicsAction`: `String sortedBy = request.getParameter("sortedBy");...`
- `topics.jsp`: fejlécek linkeké alakítása
- `PostDAO.getTopics()` paraméterezése `sortedBy`-al és `sortDirection`-el.

- A `topics.jsp`-ben tegyük lehetővé a keresést, "subject" (témakör) alapján.
- Ha találat van, akkor ugorjunk az illető topic-ra, ha nincs akkor jelenítsünk meg megfelelő hibaüzenetet.

módosítások:

- `SearchTopicAction` létrehozása
- `SearchForm` létrehozása
- `strutsConfig.xml`: form és action
- `PostDAO` `public Long getTopic(String subject)`
- `ErrorConstants`: `public static final String TOPIC_NOT_FOUND = "webforum.topicnotfound";`
- `GlobalErrors`: `error.search.failed=No topic with this subject can be found`
- `topics.jsp`: megfelelő form hozzáadása

- A `topics.jsb`-ben, ha a felhasználó be van jelentkezve, a `Log Out` link mellé kínáljuk fel az `Edit Profile` linket is.
- Ez a `register.jsp`-hez kell továbbítson, de ellentétben a `Register`-el előre fel kell töltsse a formot a bejelentkezett felhasználó adataival (kivéve a jelszót, mert az nem része a `User` osztálynak. Ezt minden egyes profile módosításnál meg kell majd adni).

módosítások:

- `Resources.properties`: `topics.guest.welcome.6=Edit profile`
- `strutsConfig.xml`: `action path="/Register"...` `forward` helyett most már kell egy `action RegisterLoadAction`, amely ellenőrzi, hogy be van-e jelentkezve az illető vagy sem. Ha igen, feltölti a form-ot az illető adataival.
- `RegisterSubmitAction`:
 - csak akkor nézze, hogy létezik-e már a felhasználónév, ha nincs senki bejelentkezve
 - ha nincs bejelentkezve, akkor `addUser`, ha be van jelentkezve, akkor `updateUser`
- `UserDAO`: `updateUser` metódus