Automatizálási és
Alkalmazott
Informatikai Tanszék

Dokumentáció

Önálló Laboratórium 1

2015 Tavasz

AUT konferencia

ASP.NET MVC alapú webalkalmazás

Készítette: Gyuris Bence gyuris.bence@hotmail.com MFSX9D

Konzulens: Gincsai Gábor

Tartalomjegyzék:

1. Fejlesztői dokumentáció

1.1. Alkalmazás áttekintése

Az alkalmazást egy Visual Studio 2013-mas sollution valósítja meg. A sollution gyökere az Events mappa. Ezen belül a webalkalmazás egyetlen projektben valósul meg, ez az Events mappa.

A projektet ASP.NET MVC 5 webalkalmazásként hoztam létre Individual User Accounts authentikációval. Az alkalmazás IIS7 szerveren fut, és egy MS SQL adatbázist használ adatok tárolására.

1.2. A projekt felépítése

A projekt mappái a következőket tartalmazzák:

- **App_Data**: Privát fájlokat tartalmaz, amelyek nem kérdezhetőek le HTTP get erőforrás kérésekkel. Itt vannak a jelentkezők által feltöltött fájlok.
- **App_Start**: Alkalmazás tulajdonságait beállító kód, konfigurációs céllal. BudleConfig (.js és .css fájlok), RouteConfig, Startup.Auth, stb.
- **Content**: Az oldal stílusát definiáló css fájlok helye.
- **Controllers**: Kontrollerek osztályai.
- **fonts**: fontawesome kiegészítéshez szükséges fájlok.
- **Heplers**: Segédosztályok, kontroller akcióinak megvalósítását segítik.
- **Models**: Modell osztályok (oldalak megjelenítéséhez, és input adat fogadásához).
- **Scripts**: .JS fájlok helye.
- **Views**: .cshtml fájlok, az alkalmazás megjelenítése.
- A projekt gyökerében található az EventContext.edmx fájl, amely az adatbázis adaptere. Ehhez tartozik az EventsRepository.cs fájl, amely az adapter generált osztályainak kiegészítéseit tartalmazza.

1.3. Adatbázis

1.3.1. Adatbázis kapcsolat

Az alkalmazás kapcsolatát az adatbázissal a Web.config connectionStrings elemében kell definiálni. A connectionStrings-ben két add elem található:

- DefaultConnection: Az authentikációhoz tartozó adatokat (azaz a felhasználói adatokat) tároló adatbázis kapcsolata.
- EventsEntities: Az előadások adatait tároló adatbázis kapcsolata.

A tesztkörnyezetben, ez a két adatbázis ugyanaz, egy helyen tárolom a felhasználói és az előadás adatokat, így a connection string megegyezik.

1.3.2. Felhasználói adatbázis

Az alkalmazás a beépített adatbázis alapú felhasználói azonosítást használja, a generált projekt ezzel kapcsolatos részén gyakorlatilag semmit nem változtattam. Az alkalmazás első indításakor az első regisztrációkor vagy bejelentkezési kísérletkor az alkalmazás létrehozza automatikusan a felhasználó kezeléssel kapcsolatos táblákat, a neki megadott adatbázisban.

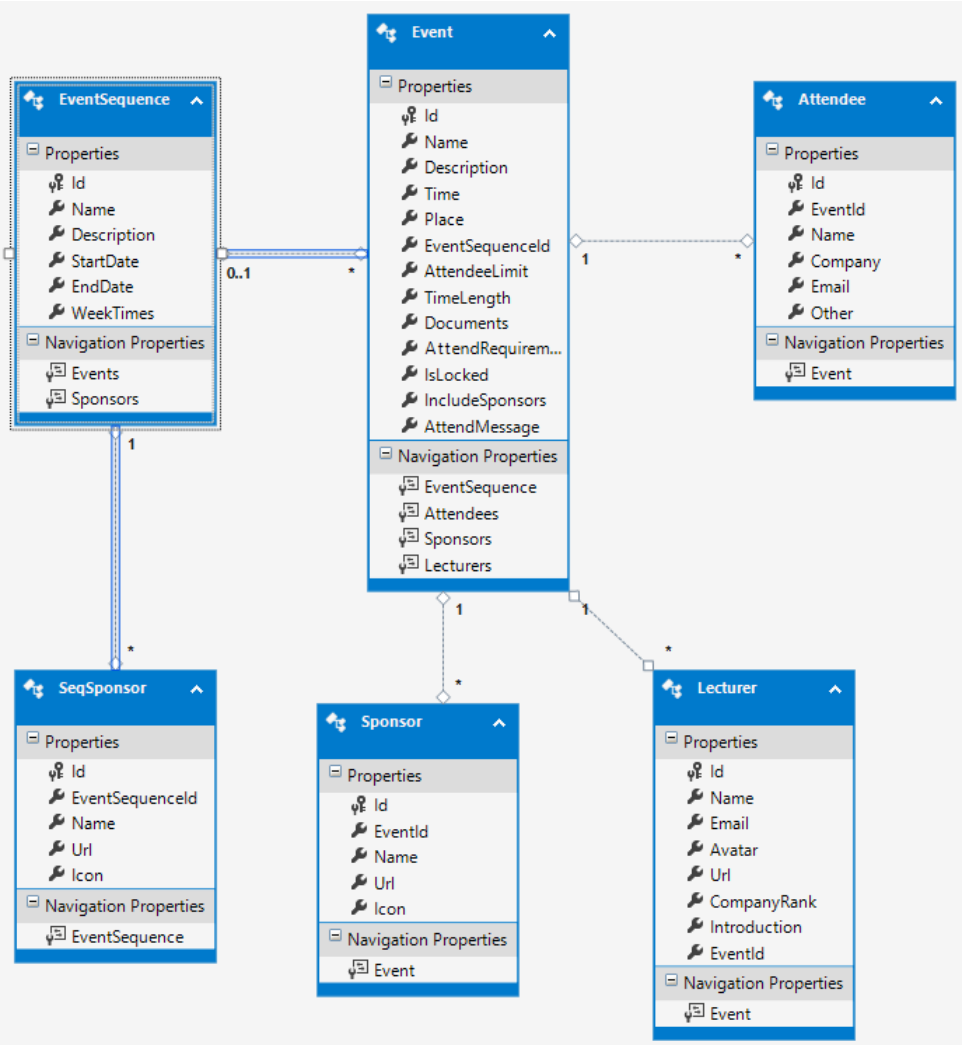
Hiba:

Nekem előfordult, hogy a táblákat, az alkalmazás valamiért rosszul generálta (rosszul nevezett el egy oszlopot) és így nem tudott működni, ezért a projektbe mellékeltem **user_db_create.sql** nevű fájlt, ezzel a szkripttel is lehet adatbázist generálni, ezzel is működni fog.

1.3.3. Tartalom adatbázis szerkezete

A portál tartalmát, az előadások adatait, az EventsEntities-ben definiált adatbázisból veszi az alkalmazás. Az adatbázis adaptert egy létező adatbázisból automatikusan generált Entity Data Model valósítja meg. Ez az EventsContext.edmx fájlban van definiálva. Ennek az adapternek az osztályait használja az alkalmazás a perzisztens adatok olvasására és írására.

Ahhoz, hogy az alkalmazás működjön létre kell hozni az itt definiált adatbázis sémát. Ezt a projekt gyökerében található **db_create.sql** szkript segítségével lehet megtenni. Az adatbázis szerkezete a következő:



Az egyes táblák és attribútumaik vázlatos leírásai:

Event: Egy előadáshoz tartozó rekord.

Name:	Előadás címe.
Description:	Előadás leírása
Time:	Előadás kezdeti időpontja.
TimeLength:	Előadás hossza, percben.
Place:	Előadás helyszíne (terem).
AttendeeLimit:	Jelentkezők maximális száma.
Documents:	Dokumentumok pontosvesszővel elválasztva, beágyazott videó HTML kódja.
AttendeeRequirements:	A jelentkezés feltételei, milyen adatokat kell megadniuk.
IsLocked:	Ha le van zárva, nem lehet jelentkezni.
IncludeSponsors:	Látszódjanak-e az előadássorozat szponzorai is.
AttendMessage:	Jelentkezéskor a jelentkező milyen szöveget lásson, utasításokat tartalmazhat.

EventSequence: Előadássorozat, névvel, leírással, kezdő- és végdátummal.

SeqSponsor: Előadássorozat szponzora, névvel, url-lel, ikonnal.

Sponsor: Előadás szponzora.

Lecturer: Előadó, névvel, email-lel, url-lel, beosztással, bemutatkozással és képpel.

Attendee: Jelentkező a hozzá tartozó esemény AttendRequirements attribútumában megadott mezőkkel. A három gyakori mezőnek külön oszlopaik vannak, a többi mező értékét besűríti az Other mezőbe.

Segédfüggvények: Az adatbázis objektumok segédfüggvényeit az EventsRepository fájlban definiáltam. A tulajdonságaikhoz kapcsolódó attribútumokat (metaadatokat) a EventsMetaData.cs fájlban deklaráltam.

FIGYELEM: Ezeket a fájlokat azért hoztam létre, hogy esetleges újrageneráláskor ne vesszenek el az általam definiált metódusok. **DE,** két helyen módosítanom kellett a generált adatokat. Sponsor.cs és SeqSponsor.cs-ben az Id-ket int? típusúvá kellett változtatnom (formból történő olvasáskor valamiért problémázott amikor nem volt nullable, de null értéket soha nem fognak felvenni).

1.4. Kontrollerek

A felhasználói műveleteket (use case-ek) a kontrollerek és az azokban található action-ök határozzák meg. Két féle művelet van. Azok ahol az action előtt szerepel az **[AllowAnonymus]** attribútum, ezeket az oldal látogatói tudják meghívni, és azok a műveletek, amelyeket csak az authentikált felhasználók tudnak meghívni, ezek az adminisztrátori műveletek. A használati eseteket a következő kontrollerek és ezek action-jei valósítják meg (mindig előre írtam az anonim engedélyezett műveleteket):

AccountController: Adminisztrátori felhasználókezelést megvalósító kontroller. Automatikusan generált. A legtöbb akciót töröltem, amit meghagytam:

Register	Új felhasználó regisztrálása. **ApplicationUserManager.IsRegistrationEnabled** statikus változó segítségével letiltható.
Login	Bejelentkezés. Ha a felhasználó nincs megerősítve (**IsEmailConfirmed**), akkor nem hajtja végre a bejelentkezést. Megerősíteni felhasználót az adatbázisban az AspNetUser tábla IsEmailConfirmed oszlopának állításával lehet.
Confirm	Adminisztrátor meg tud erősíteni egy regisztrált felhasználót ezzel az akcióval.
Delete	Adminisztrátor el tud utasítani egy felhasználót

EventController: A legnagyobb kontroller, a legtöbb kód itt van. Előadások kezeléséért felelős.

Show	Megjelenít egy előadást. *EventShowModel*-t használ a megjelenített adatok átadására.
Attend	Jelentkezés egy előadásra. Ellenőrzi hogy a jelentkezés tartalmaz e minden szükséges mezőt és engedélyezett-e a jelentkezés, ha nem hibával tér vissza. *AttendEventModel*-t használ adatok fogadására és átadására.
AttendBigFile	Egy nagy fájlt vár feltöltésre. JS kód indítja el az aszinkron feltöltést jelentkezésnél a csatolmány megadásakor, ha a fájl nagyobb, mint 2 MB, így reszponzív marad a felület. Az akció elteszi egy ideiglenes mappába a fájlt, küld egy kulcsot a user-nek, aki azzal hivatkozni tud rá, ha befejezte a jelentkezést.
Create	Új előadást hoz létre. CreateEventModel-t használ az adatok olvasására, ez tartalmazza az előadás minden adatát.
Edit	Módosít egy előadást. A .cshml oldal ugyanaz, mint a létrehozáskor.
Delete	Töröl egy előadást, és minden hozzá tartozó entitást (szponzorok, előadók, jelentkezők, fájlok).
Attends	Listázza a jelentkezőket táblázatos formában.
Export	Exportálja a jelentkezőket .xlsx fájlba. Az XlsxHelper osztályt hoztam létre az egyszerűbb kezelésért.
Upload	Oldal, ahol képeket, dokumentumokat tudunk feltölteni és videót tudunk megadni az előadáshoz.
UploadFile	Egy fájl feltöltését csinálja meg.
EmbedVideo	Videót ágyaz be az előadáshoz.
DeleteFile	Töröl egy feltöltött képet/dokumentumot.
ToggleLock	Lezárja az előadást, vagy feloldja a lezárást.

EventSequenceController: Automatikusan generált kontroller, előadássorozatok kezelésére.

Details	Egy előadássorozat részleteit jeleníti meg. Mindenki meghívhatja.
Index	Automatikusan generált, listázza az előadássorozatokat.
Create	Létrehozza az előadássorozatot. EventSequenceModel-t használ a szponzorok kezelésére.
Edit	Módosítja. Hasonlít a létrehozásra, ugyanaz a megjelenítés.
Delete	Törlés oldalt nyitja meg.
DeleteConfirmed	Törli az előadássorozatot

FileController: EventAttachment akciójával adott előadás adott nevű, jelentkező által feltöltött csatolmány fájlt lehet letölteni. Jogosultságellenőrzést valósít meg, az alkalmazás privát fájljait csak adminisztrátornak adja ki.

HomeController: Index akciója a kezdőlapot valósítja meg.

1.5. Modellek

A **Models** mappában valósítottam meg a modellek kezelésével kapcsolatos osztályokat.

Models/Event: Az általam létrehozott modell osztályokat tartalmazza. Általában az osztályok konstruktora egy entitást vár paraméterként, vagy egy függvénye visszaadja azt az entitást, így tudnak a kontrollerek adatokat fogadni a kéréstől és adatokat átadni a megjelenítésnek. Az osztályok tulajdonságainak megadtam attribútumokat, amelyek a megjelenítéssel (Display) és validációval (pl. Required) kapcsolatosak.

Models/Validation: Saját validációs attribútumokat hoztam itt létre. A PhoneNumber egy telefonszám érvényességét ellenőrzi, ezt a jelentkező form használja. A RequiredArray azt ellenőrzi, hogy egy tömb egyik eleme se legyen üres.

1.6. Alkalmazás fájlok

Az alkalmazás a perzisztens adatait az adatbázison kívül még a mappájában lévő fájlokban is tárolja. Így menti el a felhasználók és adminisztrátorok által feltöltött fájlokat.

Az adminisztrátorok által feltöltött fájlok publikusak (szponrozok ikonjai: **/Sponsors**, **/SeqSponsors**, előadók profilképei: **/Lecturers**, előadások dokumentumai: **/Events/{id}/**). Ezek a projekt gyökerében az **Uploads** mappában találhatóak meg, az oldalak közvetlenül címzik az itt található fájlokat. A címképzést az Events.Helpers.Base osztály statikus függvényei valósítják meg:

Base.Url: Az alkalmazás gyökér url-je (pl. /Events/, tesztkörnyezetben csak /).

Base.UploadsUrl: Base.Url + Uploads/

A felhasználók által feltöltött fájlok privátak. Ezek a jelentkezésekben megadott csatolmányok. Ezek a fájlok az **App_Data/Private/{eventId}** könyvtárban találhatóak. Ez a mappa nem címezhető közvetlenül, csak kontroller akcióját keresztül, ami megvalósítja a jogosultságkezelést. Ezen a mappán belül található egy **Attendees** és egy **Temp** mappa. A Temp mappában találhatóak meg az ideiglenesen feltöltött nagy fájlok.

Figyelem: A Temp mappa fájljaira nem hoztam létre semmi logikát, ami kitörölné a régen feltöltött és nem használt ideiglenes fájlokat, csak ha a felhasználó befejezi a jelentkezést, akkor törlődnek, egyébként megmaradnak.

1.7. Web.config

A web.config-ban, az adatbázis kapcsolatokon kívül, két fontos dolgot állítottam be.

Security/RequestFiltering elemben és a system.web/httpRuntime elemben a maximálisan feltölthető fájl méretét és feltöltés hosszát állítottam be 1GB-re és 1 órára.

A handlers-ben hozzáadtam egy szabályt, amely szerint ha a kérés /Files/* alakú, akkor mindenképpen egy kontroller egy akcióját rendelje a kéréshez (FileController/EventAttachment), így a privát fájlok route-ja megfelelő lesz (akkor is ha a kiterjesztés .pdf vagy .jpg stb.).

1.8. RouteConfig.cs

Itt két speciális route lekézését adtam meg. Az Admin route-hoz az Account/Login oldalt rendelem, valamint a Files/{eventId}/{fileName} oldalhoz a File/EventAttachment/eventId={}&fileName={} akciót rendelem. A kezdőoldalt is itt állítottam be Home/Index-re.

1.9. JavaScript fájlok

Három saját .js fájlt hoztam létre:

events.js: Minden oldal tartalmazza. Datepicker konfigurációt és input[type='number'] validációt tartalmaz, valamint fájl feltöltő segédfüggvényeket.

events-upload.js: Dokumentumfeltöltés jquery kódját tartalmazza. Ha egy .file-uploader elem tartalma megváltozik, feltölti a kiválasztott fájlt, és a visszadott tartalmat megjeleníti a hozzá tartozó container-ben. Az X gombokhoz eseménykezelőt rendel kattintásra. Az X gombok kattintásakor a hozzá tartozó fájl törléséhez tartozó akciót hívja meg, és eltávolítja az elemet. Az #embed-html elem tartalmának változásakor az EmbedVideo akciót hívja meg ajax hívásként a tartalmával paraméterezve.

event-create.js: Előadás és előadássorozat létrehozásához/módosításához szükséges jquery kód.

- A szponzorok és az előadók paneleit tartja karban. + gomb nyomására újat hoz létre, X gomb nyomására töröl, valamint a name attribútumok értékét állítja be.
- Tinymce-t inizializálja.
- Az előadás időpontját beállító inputok közötti ugrásokat valósítja meg.

1.10. Nuget Csomagok

Az általam letöltött csomagok:

- Font Awesome: Ikonok megjelenítésére (X, +).
- jQuery UI: datepicker használatához (UI Lightness témával).
- OpenXML SDK 2.0: xlsx excel fájlok létrehozására használom.
- TinyMCE.jQuery: Formázható szövegdobozt ad.