



**Eötvös Loránd Tudományegyetem
Informatikai Kar**

Webes alkalmazások fejlesztése

9. előadás

Webszolgáltatások felhasználása II. (ASP.NET Core)

Cserép Máté

mcserep@inf.elte.hu

<http://mcserep.web.elte.hu>

Webszolgáltatások felhasználása

Adatok titkosítása memóriában

- A teljes biztonsághoz a memóriában lévő kényes tartalmat is titkosítani kell, mivel az is potenciális támadási felület
 - lehetőség szerint csak a feldolgozás időtartamára szerepeljen titkosítatlan információ a memóriában, egyébként kódolva tároljuk
- Szövegek titkosított kezelésére szolgál a **SecureString** típus, amely alapvetően titkosítva tárolja a jelszót, és csak lekéréskor (**ToString**) dekódolja
 - a grafikus felületen a jelszavak titkosított bekérését a **PasswordBox** biztosítja, a **Password** tulajdonság feloldja a titkosítást (amely nem függőségi tulajdonság, így nem köthető)

Webszolgáltatások felhasználása

Adatok titkosítása memóriában

- Pl.:

```
<Button Content="Bejelentkezés"  
        Command="{Binding LoginCommand}"  
        CommandParameter="{Binding  
                            ElementName=passwordBox}" .../>  
  
<!-- magát a jelszóbekérőt adjuk át -->
```

...

```
LoginCommand = new DelegateCommand(param =>  
{  
    _model.Login(UserName,  
                 (param as PasswordBox).Password);  
    // kiolvassuk a titkosított jelszót  
    ...  
}
```

Webszolgáltatások felhasználása

Példa

Feladat: Valósítsuk meg az utazási ügynökség épületeit karbantartó asztali alkalmazást.

- adjunk lehetőséget képek megtekintésére, hozzáadására, törlésére
 - a képet fájlból töltjük be, majd átméretezzük (kis és nagy méretben, PNG formátumban)
 - a képeket egyedi azonosítóval látjuk el, valamint az épület azonosítójával
- a biztonság növelésére az adatkezelést autentikációhoz kötjük (*ASP.NET Core Identity* segítségével), így a felhasználónak előbb be kell jelentkezniük az alkalmazásba

Webszolgáltatások felhasználása

Példa

Tervezés:

- létrehozunk egy vezérlőt (**BuildingImageController**), valamint egy adatátviteli típust (**ImageDTO**) a képkezeléshez
- a képeket alapvetően byte tömbként kezeljük, a szolgáltatás nem is ismeri azok képi tartalmát
- a képbetöltést egy segédtípusban (**ImageHandler**) végezzük
- a képek megjelenítéséhez átalakítást végzünk (**BuildingImageConverter**), ami **BitmapImage** típusra alakítja a tömböt, ezeket Image vezérlővel jelenítjük meg
- külön nézetbe szervezzük az épület adatainak megadását (**BuldingEditorWindow**)

Webszolgáltatások felhasználása

Példa

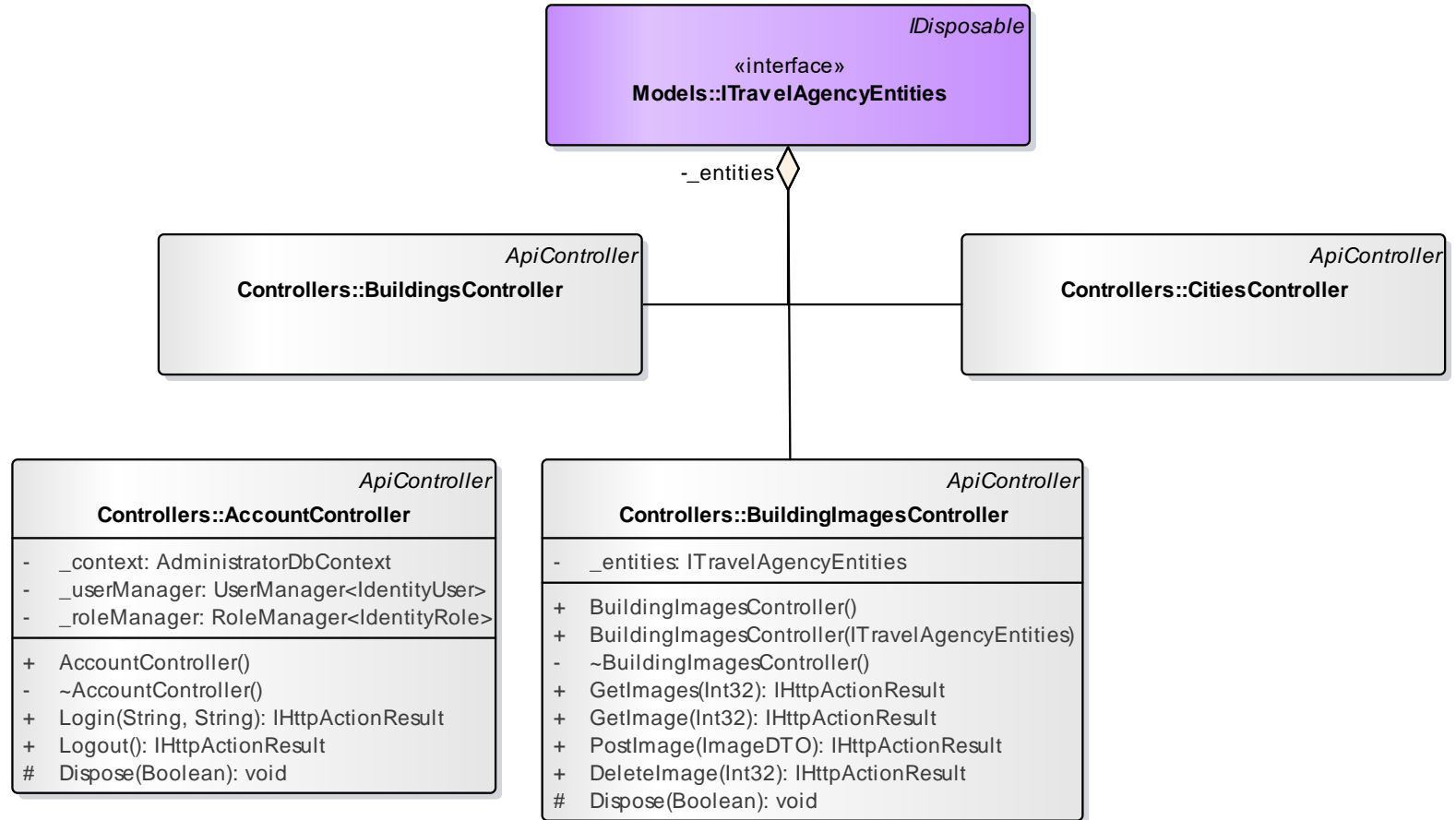
Tervezés:

- létrehozunk egy vezérlőt a felhasználó-kezeléshez (**AccountController**), ebben lehetőséget adunk bejelentkezésre (**Login**) és kijelentkezésre (**Logout**)
- a szolgáltatásban attribútum (**Authorize**) segítségével korlátozzuk az akciófüggvényekhez való hozzáférést (csak a rendszergazda csoportban lévő felhasználókra)
- kliens oldalon megjelenik a két új művelet a modellben (**LoginAsync**, **LogoutAsync**)
- a bejelentkezéshez egy külön nézetet (**LoginWindow**), valamint nézetmodellt (**LoginViewModel**) hozunk létre

Webszolgáltatások felhasználása

Példa

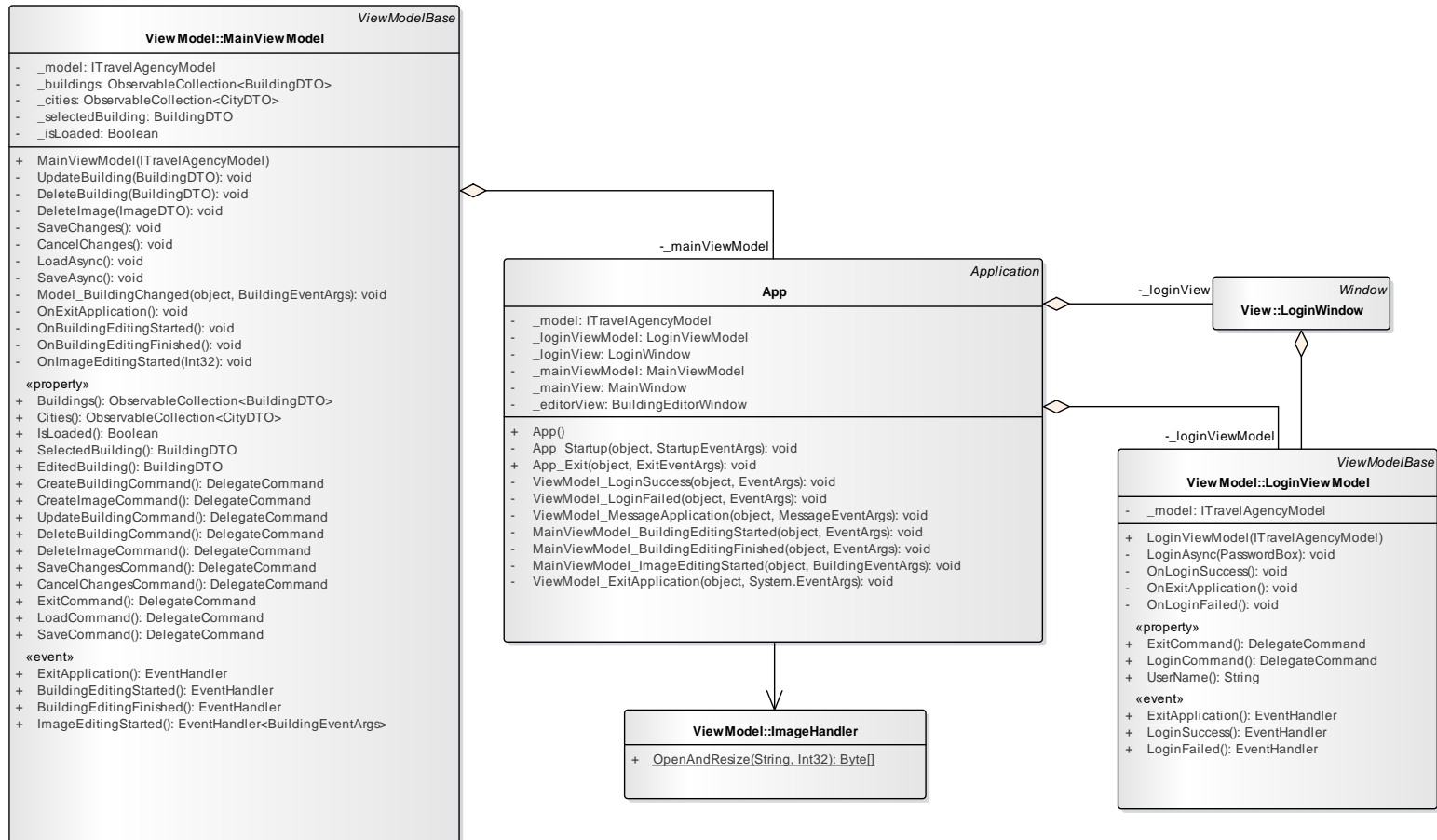
Tervezés (szolgáltatás):



Webszolgáltatások felhasználása

Példa

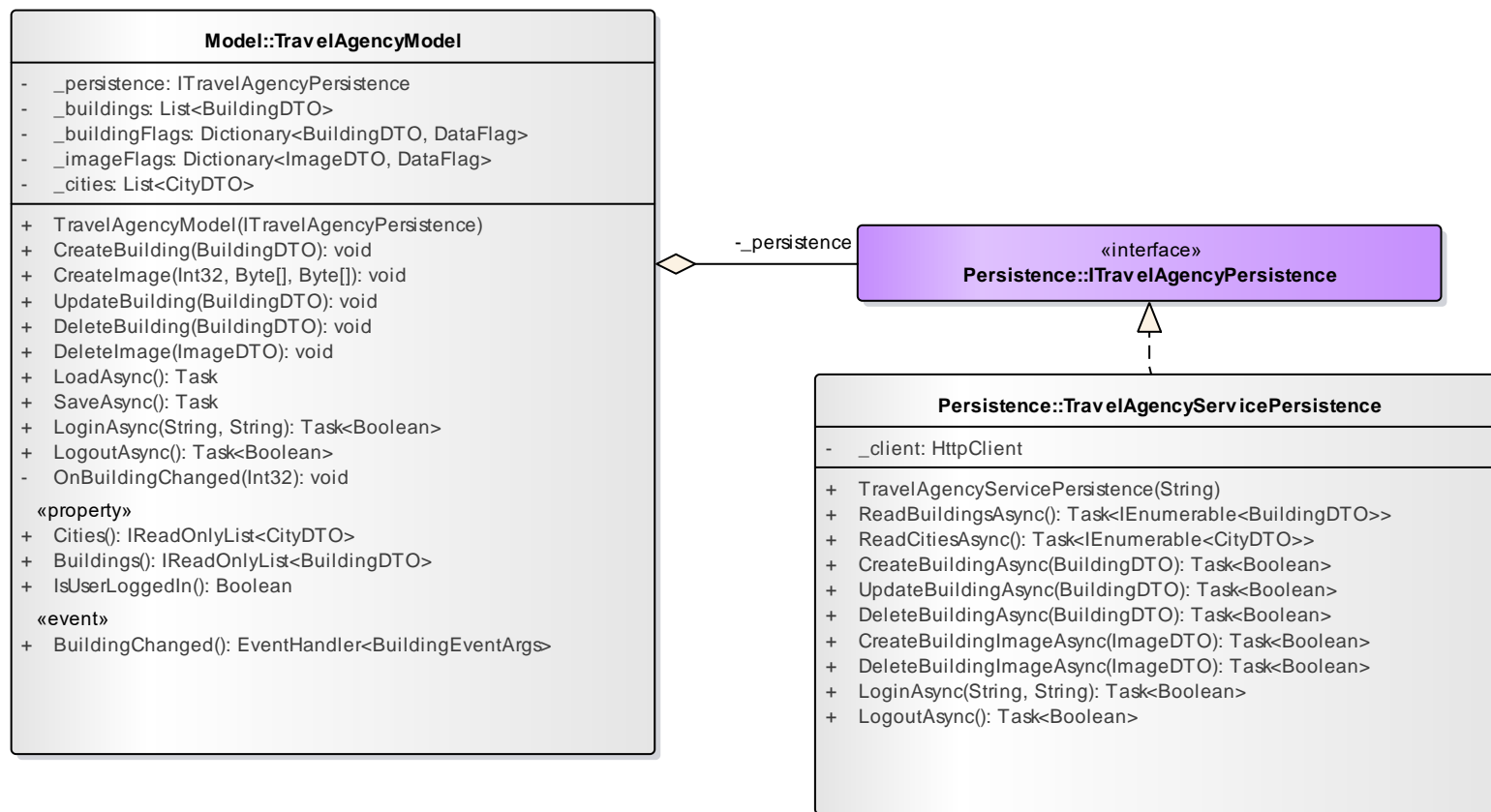
Tervezés (kliens):



Webszolgáltatások felhasználása

Példa

Tervezés (kliens):



Webszolgáltatások felhasználása

Példa

Megvalósítás (BuildingImagesController.cs):

```
[Authorize(Roles = "administrator")]
    // csak bejelentkezett adminisztrátoroknak
public IActionResult PostImage([FromBody]
    ImageDTO image)
{
    BuildingImage buildingImage = new ...
    ...
    _context.SaveChanges();
    return CreatedAtAction(nameof(GetImage),
        new { id = buildingImage.Id },
        buildingImage.Id);
    // csak az azonosítót küldjük vissza
    ...
}
```

Webszolgáltatások felhasználása

Példa

Megvalósítás (ImageHandler.cs):

```
public static Byte[] OpenAndResize(String path,
                                    Int32 height)
{
    BitmapImage image = new BitmapImage();
    // kép betöltése
    image.BeginInit();
    image.UriSource = new Uri(path);
    image.DecodePixelHeight = height;
    // megadott méretre
    image.EndInit();
}
```

Webszolgáltatások felhasználása

Példa

Megvalósítás (ImageHandler.cs):

```
PngBitmapEncoder encoder =  
    new PngBitmapEncoder();  
    // átalakítás PNG formátumra  
encoder.Frames.Add(BitmapFrame.Create(image));  
  
using (MemoryStream stream =  
    new MemoryStream())  
    // átalakítás byte-tömbre  
    {  
        encoder.Save(stream);  
        return stream.ToArray();  
    }  
}
```

Webszolgáltatások felhasználása

Kleins alkalmazások

- A webszolgáltatások a szerver-oldali backend alkalmazás általános körű felhasználását teszik lehetővé.
- Tetszőleges platformú kliens alkalmazás nyújthat kliens oldali felületet a webszolgáltatáshoz.
 - Asztali alkalmazás, mobil alkalmazás, weboldal
 - Más webszolgáltatás is felhasználhatja
- A backend webszolgáltatás és a frontend kliensek eltérő platformokon és programozási nyelveken, eltérő fejlesztő csapatok által, párhuzamosan is fejleszthetőek.

Webszolgáltatások felhasználása

Példa

Feladat: Valósítsuk meg egy utazási ügynökség weblapját, amelyben apartmanok között böngészhetünk.

- könnyítsük meg a foglalást az által, hogy a dátumot egy naptár (`glDatePicker`) segítségével lehessen kiválasztani
- a dátumkiválasztó automatikusan lekérdezi (jQuery segítségével) a szolgáltatástól, mely napok szabadok
- a szolgáltatása vezérlője (`RentDateController`) egy műveletet biztosít (`Get`) a megfelelő napok lekérdezésére
- magát a lekérdezést a modell (`TravelService`) biztosítja (`GetRentDates`), amely adott egy hónap körüli

Webszolgáltatások felhasználása

Példa

Megvalósítás (RentDateController.cs):

```
[ApiController]
[RoutePrefix("api/[controller]")]
// útvonal feloldás megadása
public class RentDateController : Controller
{
    ...
    [Route("{apartmentId}/{year}/{month}")]
    // útvonal feloldás megadása
    public IEnumerable<DateTime> Get(
        Int32? apartmentId,
        Int32? year, Int32? month)
    { ... }
}
```

Webszolgáltatások felhasználása

Példa

Megvalósítás (Rent/Index.cshtml):

```
<script type="text/javascript">
    $(window).load(function () {
        ...
        jQuery.getJSON("api/rentdate/" +
            @Model.Apartment.Id + "/" + year + "/" +
            month,
            function (data)
            {
                options.selectableDates =
                    parseDates(data);
            });
        ...
    }
}
```