Alkalmazáséletciklus-menedzsment

Microsoft Azure

2016.

# Bevezetés

## Célkitűzés

A labor során a hallgatók megismerkednek a TFVC verziókezelő használatával, a VSTS kódkezelő és Team Build funkcióival, valamint bepillantást nyernek a munkaszervezési funkcióiba is.

## Előfeltételek

A labor elvégzéséhez szükséges eszközök:

* Visual Studio 2015, (Web development tools)
* Azure előfizetés
* Microsoft account
* A fenti komponensek telepítéséhez rendszergazdai jogosultság szükséges, a használatukhoz viszont nem

## Amit érdemes átnézned

* Kapcsolódó előadásanyag

## Oktatói útmutatás

A laboranyag gyakorlatilag nem tartalmaz kódolási feladatot, alapvetően különböző weboldalakon, illetve Visual Studioban végrehajtandó konfigurálási műveleteket hajtunk végre. Ebből következően sokkal könnyebb lemaradni, figyeljünk rá, hogy ne ugráljunk túl gyorsan a felületek között, illetve gyakran zárkóztassuk fel az esetleges lemaradókat.

# Team projekt létrehozása

Alakítsunk 2-5 fős csoportokat, minden csoportban jelöljünk ki valakit, aki a csoport tulajdonosa lesz. A laborvezető mindenképp legyen tulajdonos.

1. Minden tulajdonos nyissa meg a <http://www.visualstudio.com> oldalt, majd válasszuk a Visual Studio Team Services részén a „Get started for free” opciót. Ezután be kell jelentkezni egy Microsoft fiókkal, majd egy URL előtagot kell választani, végül forráskódkezelőt.
2. Az URL előtag bármi lehet, lehetőleg könnyen megjegyezhető legyen, a forráskódkezelőnek válasszunk TFVC-t.
3. Némi szöszölés után már egy létrehozott Team Project áttekintő nézetét látjuk, váltsunk vissza a bal felső sarokban a VSTS fiók áttekintő nézetére (ennek webcíme megegyezik a választott VSTS URL-lel).
4. A Users fülön adjuk hozzá a többi csapattag Microsoft fiókját. A többi csapattag nyissa meg a VSTS URL-t, lépjen be a saját Microsoft account-jával: a tulajdonoséhoz nagyon hasonló áttekintő felülethez jutnak.
5. A tulajdonosok a Recent projects & teams részén válasszák a New opciót. Az új projekt adatai a következők legyenek:

* Név – SuperWebApp
* Process Template – Scrum
* Version control – TFVC

Az előadásanyag alapján mondhatunk néhány mondatot a többi opcióról is.

1. Némi szöszölés után az új projekt áttekintő nézetére jutunk. A csapattagok nem látják alapból az új projektet, előbb hozzá kell rendelni őket, de ezt pont meg tudjuk tenni a projekt áttekintő nézetén. A csoport tulajdonosok kattintsanak a Team Members részen az Invite gombra, a megjelenő felületen pedig adjuk hozzá a csapattagokat.
2. A csapattagok ezután a Recent projects & teams részen látják a projektet. Ha nem jelenne meg, a Browse linkre kattintva válasszuk ki a team projektet, majd a Navigate gombra nyomva nekik is előjön a projekt áttekintő nézete.

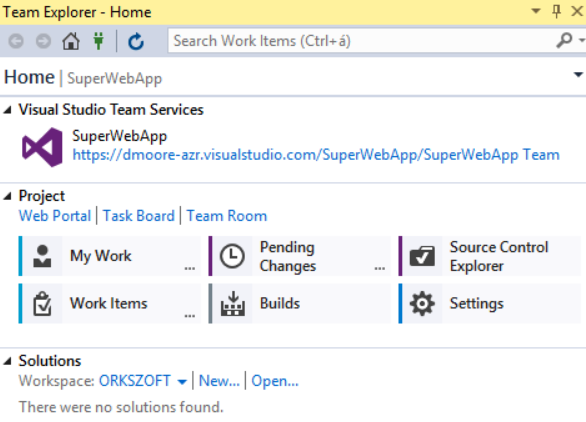
Ebben az állapotban tehát mindenki ugyanannak a Team Projektnek az áttekintő nézetét látja.

# Team Explorer, alkalmazás létrehozása

Most már létrejött a felhőben a Team Projektünk. Egyelőre kód szempontjából (is) elég üres, ezt a Code fülön nézhetjük meg: alapból a $/<projektnév> elérési útvonalon állunk, a jobb oldali rész üres (leszámítva egy BuildProcessTemplates mappát). A $ kezdetű útvonalakat szerveroldali elérési útvonalnak hívjuk.

A feladatunk, hogy leképezzük a szerver oldali kódtárat egy a lokális gépen található sima könyvtárnak.

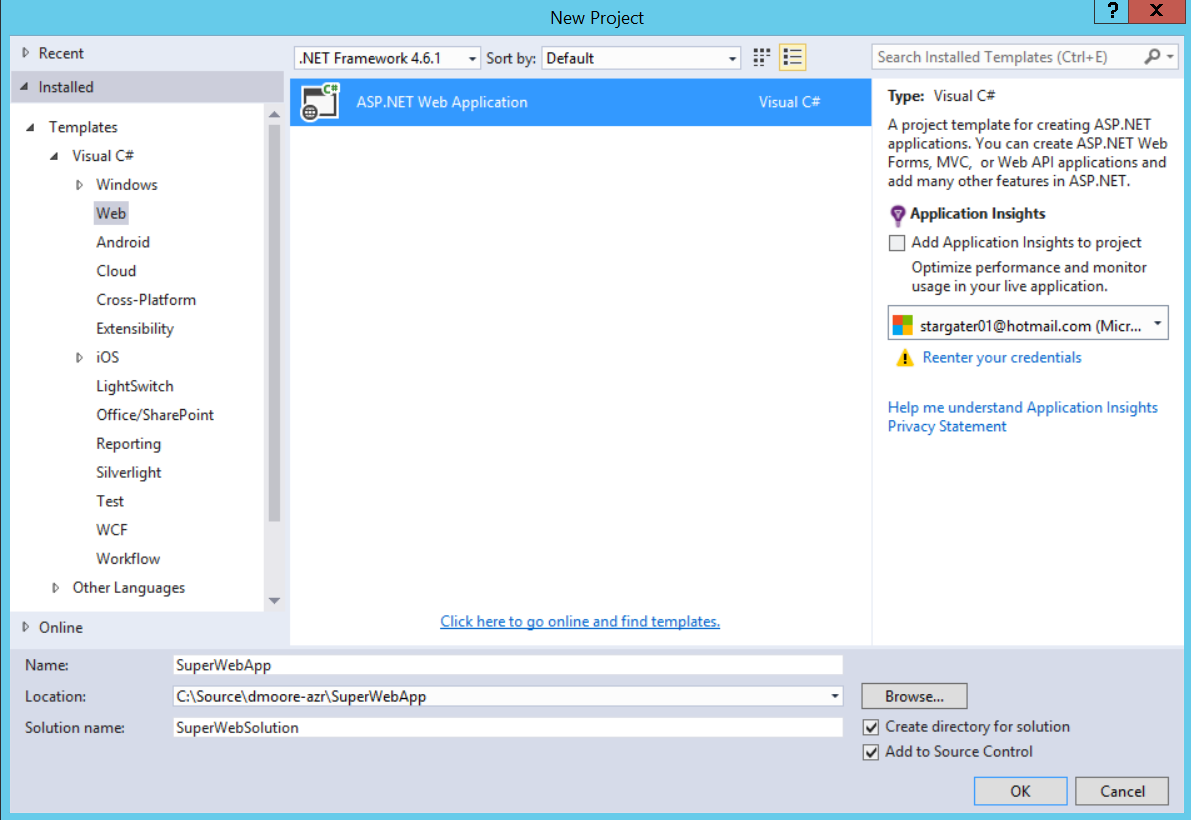
1. A projekt áttekintő nézetén válasszuk az Open in Visual Studio opciót. Ez egy olyan URL (vsweb://), amit a Visual Studio értelmezni tud – ha a böngésző megkérdezi, engedélyezzük, hogy a Visual Studio kaphassa meg a linket.
2. Ezek után a Visual Studio megnézi, hogy az a projekt szerver oldali útvonala le van-e már képezve. Valószínűleg nincs, úgyhogy egy könyvtárat kell megadnunk (lehet még nem létező is). Ez lesz a $/<projektnév> lokális leképezése. A Map&Get gombra nyomva kész is a leképezés, létrejön egy új TFS workspace (lásd lentebb)
3. Mutassuk meg a Team Explorer ablakot (a View menüből is előhozható). A felső részen lévő link, illetve a Web Portal link is a Team Projekt áttekintő részére visz.



A Task Board és a Team Room a megfelelő projekt portál részre visznek. A Project rész gombjai különböző Visual Studio-n belüli felületeket aktiválnak, néhányukkal foglalkozunk még később, most koncentráljunk a Solutions részre. A Workspace: felirat után találjuk a workspace-ünket. A workspace fogja össze többek között a leképezéseinket. Mivel munkaállomáshoz kötött, ezért általában a gépünk neve lesz a workspace neve.

1. Klikkeljünk a kis nyílra a workspace neve mellett, majd válasszuk a Manage Workspaces… opciót. Egy listát látunk a workspace-ekkel, valószínűleg csak egy elem van a listában, válasszuk ki, majd Edit gomb. Nézzük meg a beállított leképezéseinket (elvileg csak egy van – amit most adtunk meg). Ha minden oké, ne változtassunk zárjuk be a dialógusablakokat, térjünk vissza a Team Explorerhez.
2. A következő műveleteket csak az egyik csdapattag hajtsa végre. A Solutions részen válasszuk a New… opciót, ez a normál Visual Studio projektvarázslót hozza fel, csak a projekt helyét eleve kitölti a workspace lokális könyvtárával.

* a projekt típusa legyen C#/Web/ASP.NET Web Application
* a projekt neve legyen SuperWebApp
* a solutioné SuperWebSolution
* Application Insights támogatás nem kell - ne klikklejük be
* legyen külön könyvtára a solution-nek – klikkeljük be
* legyen verziókezelő támogatás – klikkeljük be



Válasszunk MVC projektet a következő beállításokkal:

* ne legyen authentikáció – Change Authentication > No Authentication
* egyelőre ne legyen Azure Host – Host in the cloud ne legyen bepipálva
* jöjjön létre unit test projekt is – Add unit tests legyen bepipálva

1. Próbáljuk ki a létrejött webes projektet. Indítsuk el. A megnyíló weboldalon nyissunk meg párat a felső sor menüpontjai közül (Home, About, Contact)
2. A unit teszteket is próbáljuk ki. Test > Run > All Tests. A teszteredményeket a Test Explorer-ben ellenőrizzük le (Test > Windows > Test Explorer)

Ne mutassuk be részletesen a generált kódot, nem a webes fejlesztés vagy a tesztírás a labor célja.

# TFVC alapok

Van tehát egy forduló-működő kezdeti verziónk, ezt már érdemes a verziókezelőbe feltölteni, megosztani a csapattagokkal.

Figyeljük meg a Solution Explorer-ben a projekt fájljai mellett a kis zöld + jeleket. Ez mutatja, hogy az adott fájl még nincs feltöltve a verziókezelőbe. A Solution Explorer-ben bármilyen szinten (fájl, könyvtár, projekt, solution) a jobbklikkel menüből kiadhatjuk a Check in… (vagy Source Control > Check In...) parancsot, ami a Team Explorer Pending Changes aloldalát aktiválja. Az Included Changes és az Excluded Changes aszerint töltődik fel, hogy a Solution Explorer-ben mit választottunk (pl. ha csak a unit teszt projektet választottuk, akkor csak az kerül az Included Changes részbe, a webes projekt az Excluded Changes-be kerül). A következő műveleteket csak a solution-t létrehozó csapattag tudja végrehajtani.

1. Vagy a solution jobklikkes menüjéből adjuk ki a Check In utasítást, vagy mutassuk meg, hogy a Pending Changes ablakban hogyan lehet az Excluded Changes elemeit a jobbklikkes menü (Include) segítségével az Included Changes-be áthelyezni.
2. Miután minden elem az Included Changes részen van, töltsük ki a check-in commentet (pl. project generated), majd nyomjunk a check-in gombra.
3. Check-in után elképzelhető, hogy a packages könyvtárat is felajánlja feltöltésre. Jobbklikk az Included Changes ablakban a packages könyvtárra > Undo…

Általában csak forráskódot és automatikusan nem előállítható asset-eket szoktunk forráskódkezelőbe tölteni.

Kiürült a Pending Changes ablak, a Solution Explorer-ben elvileg minden elem mellett kis lakat ikon mutatja, hogy az adott elem a verziókezelőben is megvan.

Itt az ideje, hogy a csapat többi tagja is eljusson idáig. A következő műveleteket ők hajtsák végre.

1. A Team Explorer-ben válasszuk a Source Control Explorer opciót.
2. A bal oldali, könyvtárakat listázó részen válasszuk a Team Projekt könyvtárát, a jobbklikkes menüben pedig a Get Latest Version opciót. Ezzel az előbb feltöltött fájlok letöltődnek a lokális mappába.

Figyeljük meg, hogy a Source Control Explorer minden szerver oldali mappára kiírja, hogy le van-e képezve, és ha igen, milyen lokális mappára.

1. A Team Explorer-ben a Solutions részen az Open… opcióval nyissuk meg a letöltött solution fájlt. Próbáljuk ki, hogy ugyanolyan jól működik-e, mint a létrehozónak.

Ebben az állapotban minden csapattagnak meg kell legyen a működő projekt. Kipróbálunk egy módosítási folyamatot.

1. Minden csapattag módosítsa ugyanazt a fájlt, pl. Webes projekt > Views > Shared > \_Layout.cshtml-ben a <title> tag tartalmát. Minden csapattag valami általa választott értéket írjon be, hogy minden módosítás más legyen. Figyeljük meg, hogy a megváltozott fájl előtti ikon – piros pipa jelzi, hogy az adott fájlban nem feltöltött változások találhatók.
2. Mentük el, próbáljuk ki, hogy lokálisan valóban átíródott-e a felirat.
3. Jobklikk a Solution Explorer-ben a módosított fájlon (vagy az egész projekten/solution-ön) > Check-in. Mindenki próbálja feltölteni a változtatását.
4. A később check-inelők conflict üzenetet kapnak és a Resolve Conflicts ablakhoz jutnak. Ők próbálják ki a „Merge Changes in Merge Tool” opciót. Mutassuk be a merge tool-t, beszéljünk a különböző feloldási stratégiáról. Bármelyik feloldási módot választhatják, majd próbálják újra a check-int. Újabb ütöközés lehetséges, ha közben más csapattagok check-ineltek. Figyeljük meg, hogy sikeres check-in esetén a módosított fájl előtti ikon újra a lakat ikon, jelezve, hogy a fájl minden változása feltöltésre került.
5. Miután mindenki sikeresen check-inelt, mindenki kérje le a legfrissebb verziót. (Solution-ön jobklikk > Get Latest Version). Nézzük meg, kié lett a végső változat.

# VSTS képességek

Röviden mutassuk be az alábbi VSTS aloldalakat:

* Code fül – Explorer
  + egy fájl szerkesztése
  + History fül
* Code fül – Changesets
* Work fül - Backlogs
  + PBI felvétele
  + PBI beütemezése sprinthez (Drag&Drop)
  + Sprint board megtekintése

# Team Build

A következő lépéseket a team projekt tulajdonosa hajtsa végre.

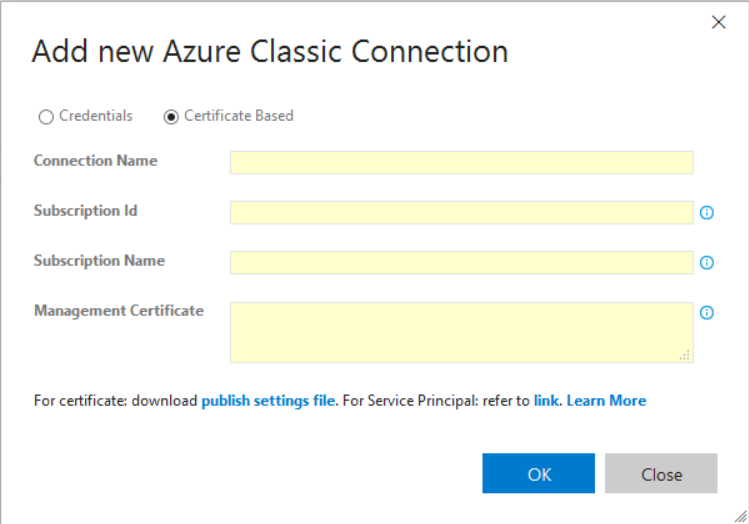
1. A Build fülön adjunk hozzá egy új fordításdefiníciót (Build Definition) az alábbi beállításokkal:

* template – Deployment fül\ Azure Web App
* forrás – a team projektünk
* repository - $/<team projekt neve>
* Default agent queue – Hosted
* Continous Integration - pipa

1. Röviden mutassuk be az egyes lépéseket:

* Nuget restore – Nuget komponensek letöltése, telepítése
* Build solution – solution fordítása. Figyeljük meg, hogy az MSBuild argumentumai úgy vannak megadva, hogy a fordítás után elkészüljön a Web Deployment Package[[1]](#footnote-2) telepítőcsomag
* Test assemblies – tesztek futtatása
* Azure Deployment – telepítés Azure Web App-ba
* Publish symbols path – debug szimbólumok publikálása
* Publish: drop a fordítás kimenetének publikálása (file share-re vagy szerverre)

1. Konfiguráljuk az Azure Deployment-et. Ehhez előbb az Azure hozzáférést kell konfigurálni. Válasszuk ki az Azure Deployment Step-et. Az Azure Subscription sorában válasszuk a Manage opciót. A team projekt adminisztrációs felületének Services fülére jutunk.
2. Az Endpoints alfülön adjunk hozzá új Azure Classic Service Endpoint-ot (New Service Endpoint gomb). A megjelenő dialógusablakban válasszuk a Certificate Based opciót. Az űrlap alatt nyissuk meg a publish settings file feliratú linket. Ezen a linken egy publishsettings kiterjesztésű állományt tölthetünk le.



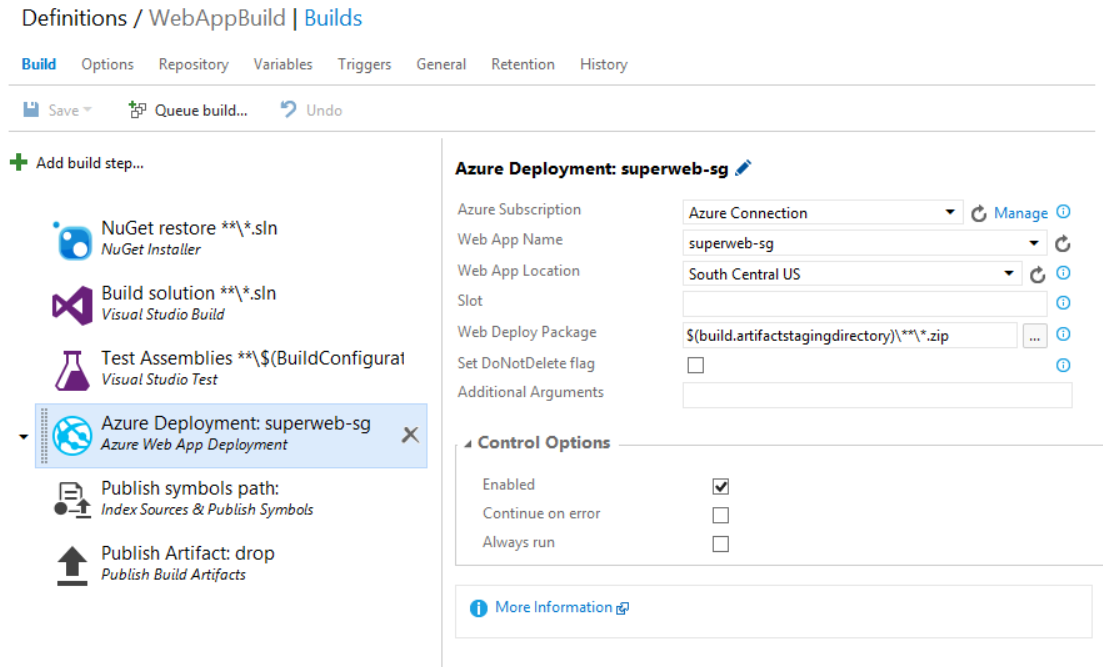
1. A letöltött fájlt nyissuk meg bármilyen szövegszerkesztőben. Láthatjuk, hogy a fájl XML formátumú. Az XML <Subscription> tag-je alapján töltsük ki a kapcsolat űrlapját.

* Connection Name – szabadon kitölthetjük (pl. Azure Connection)
* Subscription Id – Subscription tag Id attribútuma
* Subscription Name – Subscription tag Name attribútuma
* Management Ceritificate – Subscription tag ManagementCertificate attribútuma

1. Az OK gomb lenyomása után létrejön az új kapcsolat, amit már a deployment folyamatban felhasználhatunk. Térjünk vissza az Azure Deployment lépés konfigurálásához.
2. Adjuk meg az Azure telepítéshez szükséges adatokat.

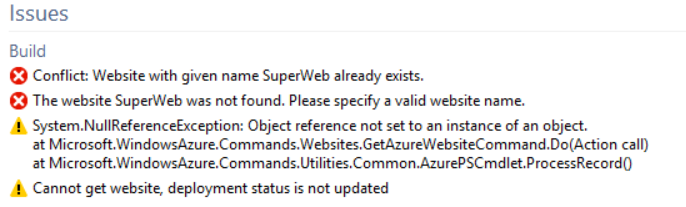
* Azure Subscription – legördülő menüből válasszuk az imént felvet kapcsolatot
* Web App Name – adjunk meg egy egyedi nevet. Fontos, hogy két csapat se válassza ugyanazt. Javaslat: <projekt rövid neve>-<csapat egyedi azonosítója>
* Web App Location – szabadon választható (lehetőleg minél közelebbi legyen)

A webalkalmazás nevének megadhatunk még nem létező nevet is, ilyenkor a deployment részeként létrejön az új webapp a megadott névvel.



1. Mentsük el a definíciót (Save gomb)
2. Indítsunk új fordítást (Queue build… gomb). Kövessük végig a build folyamatot.
3. A folyamat vége után a Build fülön a definíciót kiválasztva a lezárult folyamatokat láthatjuk. Nyissuk meg az előbb véget ért folyamat összefoglalóját (duplaklikk a folyamaton). Ellenőrizzük a teszteredményeket, töltsük le a drop eredményét (Artifacts menüpont) – a weboldal telepítőcsomagját tartalmazza. A bal oldali fából egy fázist választva az adott fázis részletes naplója jelenik meg – ezt is kipróbálhatjuk. Végül a http://<Web App Name>.azurewebsites.net címen ellenőrizzük, hogy működik-e a weboldalunk.

Gyakori hiba, hogy mégsem elég egyedi a webapp neve, ekkor kaphatunk ellentmondásosnak tűnő hibaüzenetet (nem létezik a webapp, de létrehozáskor mégis „már van ugyanilyen nevű” hiba fogad)



Próbáljunk más, más/egyedibb nevet megadni az Azure Deployment lépés Web App Name beállításánál.

1. A <http://portal.azure.com> oldalon is ellenőrizhetjük, hogy tényleg létrejött az új web app.
2. Csapatonként egy tag hajtsa végre a következőket. Módosítsunk valamit a webalkalmazásban, pl. a \_Layout.cshtml-ben a <title> tag-ben, vagy például a Html.ActionLink("Application name",…) függvényhívásban az első paramétert írjuk át. Lokálisan ellenőrizzük a módosítást (Futtatás – F5), majd töltsük föl a változtatást a korábban ismertetett módon.
3. A team projektünk weboldalán a Build fülön ellenőrizzük, hogy elindult-e automatikusan a Build folyamat. (Válasszuk ki a fordítási definíciót, majd a jobboldali lista fölött válasszuk a „Queued” szűrőt). Most is kövessük végig a fordítási folyamatot. A végén ellenőrizzük, hogy az Azure-ba telepített oldal is megváltozott-e.
4. Mutassuk meg a fordítási definíció Triggers fülén a Continuous Integration jelölőnégyzetet, ami engedélyezi a fordítás triggerelését minden check-inre. A folyamat létrehozásakor a varázslóban adtuk meg (lásd előbb).

A Team Explorer Builds gombját megnyomva is listázhatjuk a team build folyamatokat. A folyamat áttekintő nézete már a megszokott webes felületen jelenik meg.

1. https://msdn.microsoft.com/en-us/library/dd394698%28v=vs.110%29.aspx [↑](#footnote-ref-2)