COMP 90018 Mobile Computing Systems Programming

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au

Welcome!

Outcomes of this tutorial:

1. Build Azure Mobile App BackEnd

1. And this is actually a .NET project

How to create a BackEnd?

Microsoft wrote this tutorial. Read and follow carefully. It has some misleading content.

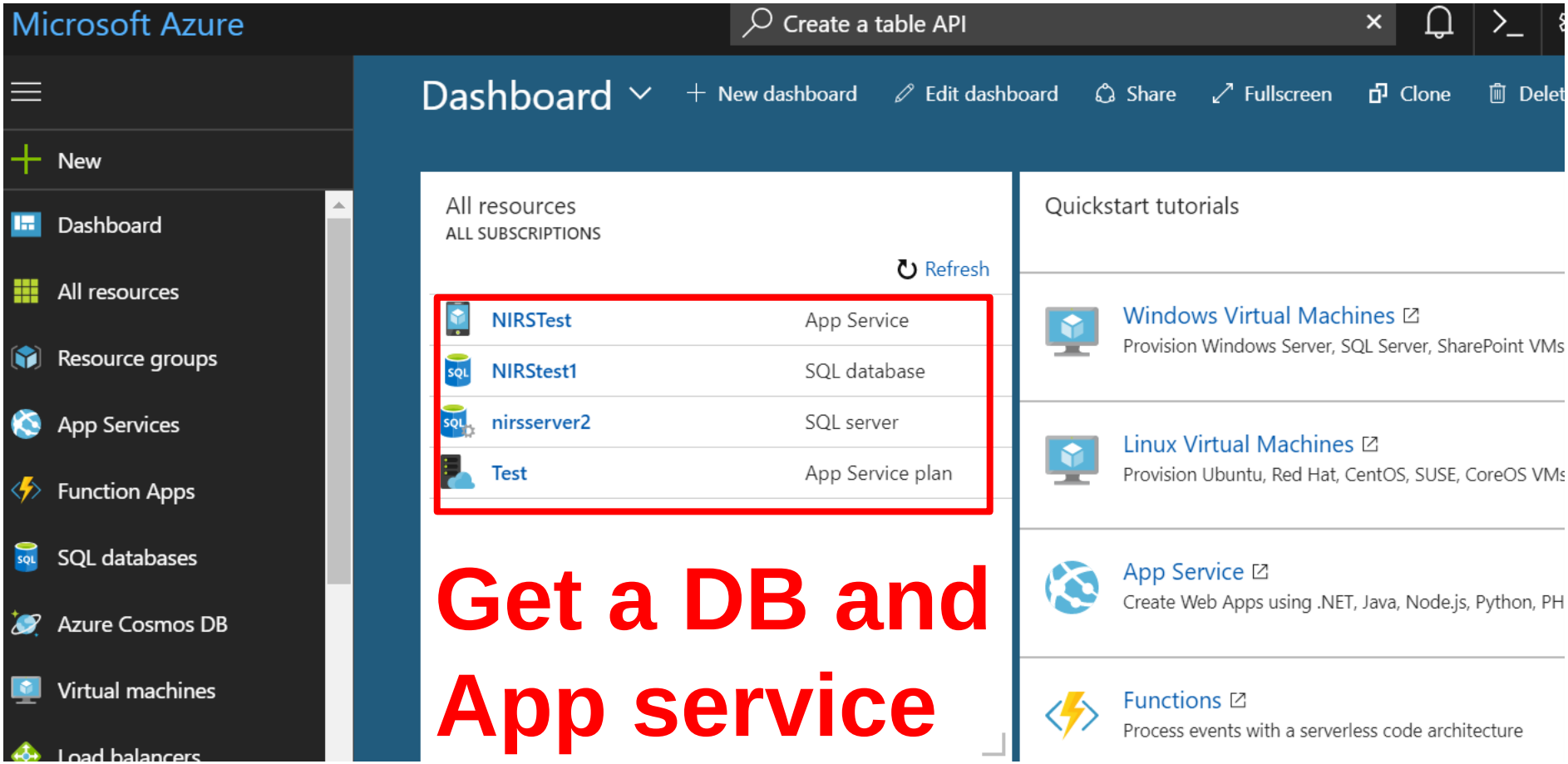
https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-dotnet-backend-how-to-use-server-sdk

This tutorial is mostly based on it.

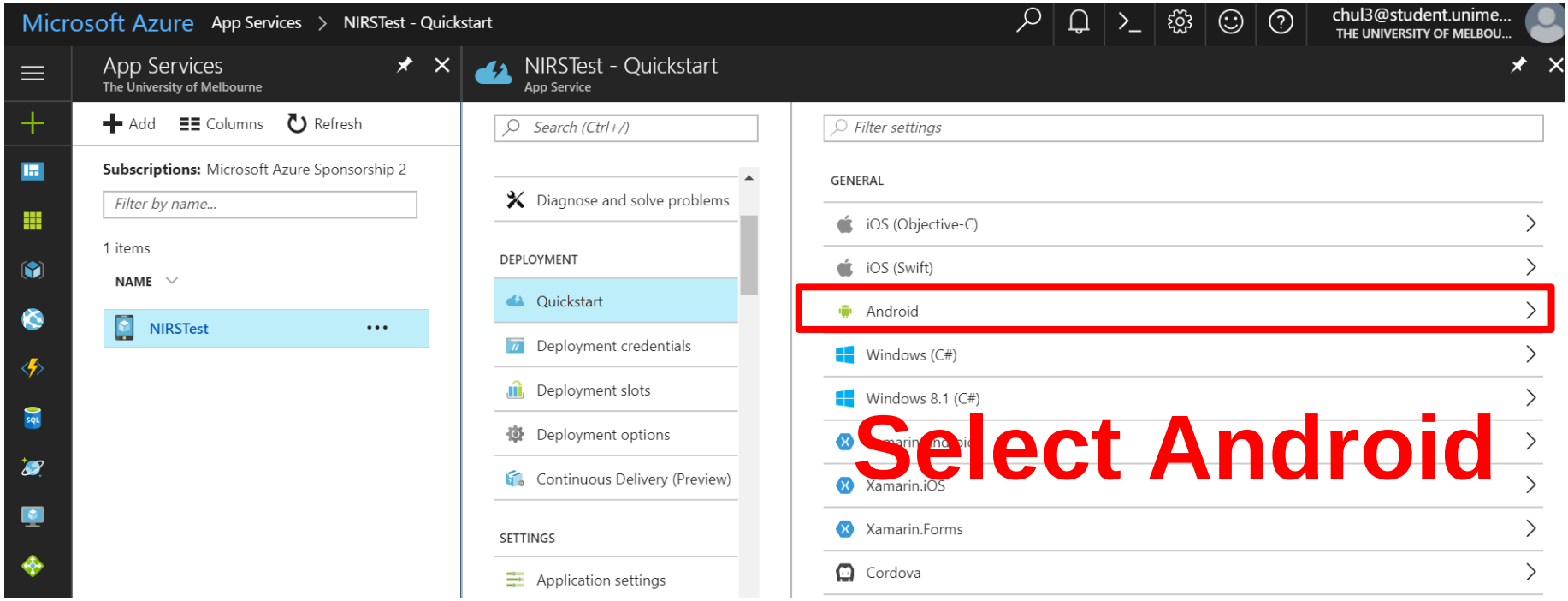
https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-dotnet-backend-how-to-use-server-sdk

First, go to Azure portal

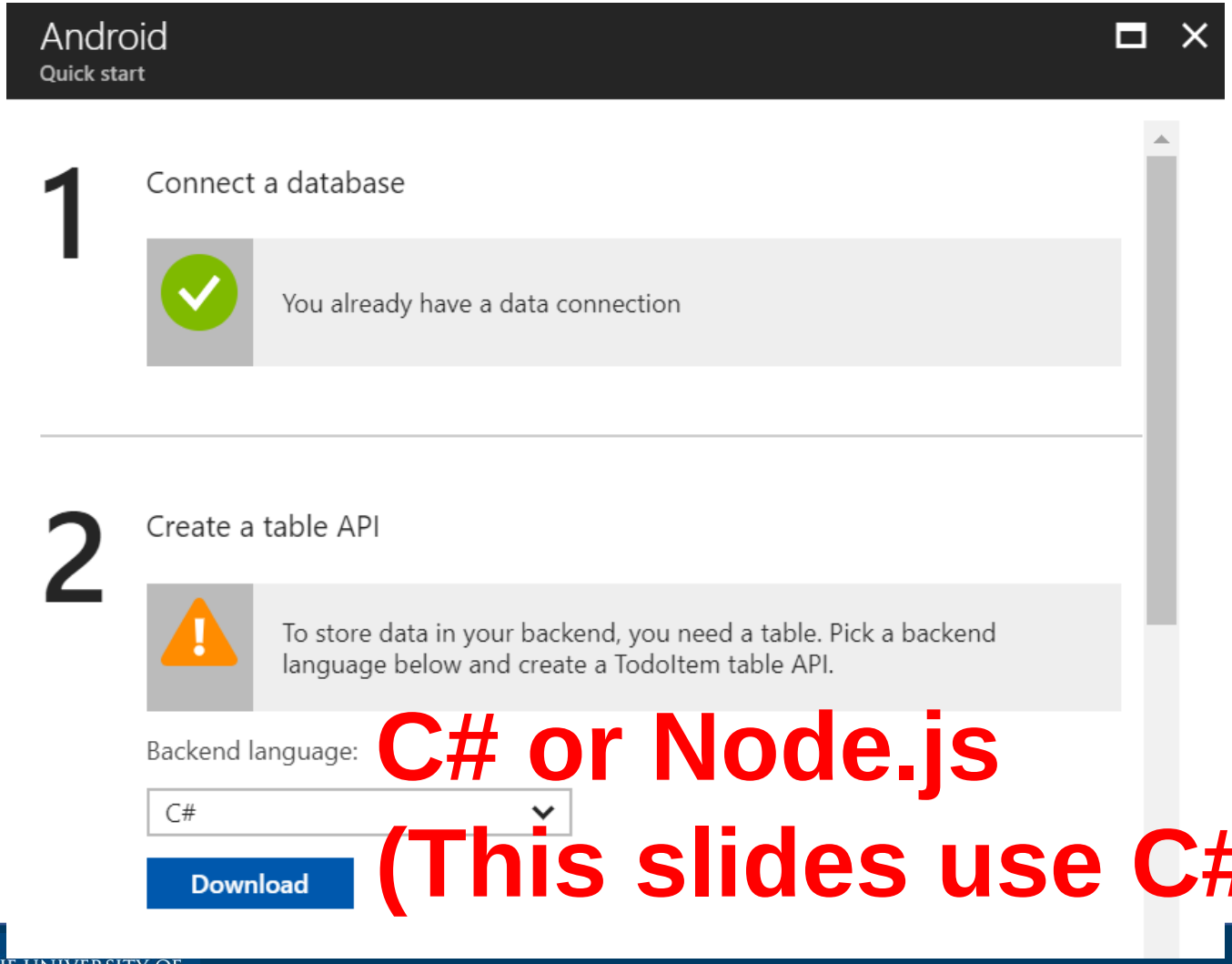
App Services BackEnd



App Services BackEnd

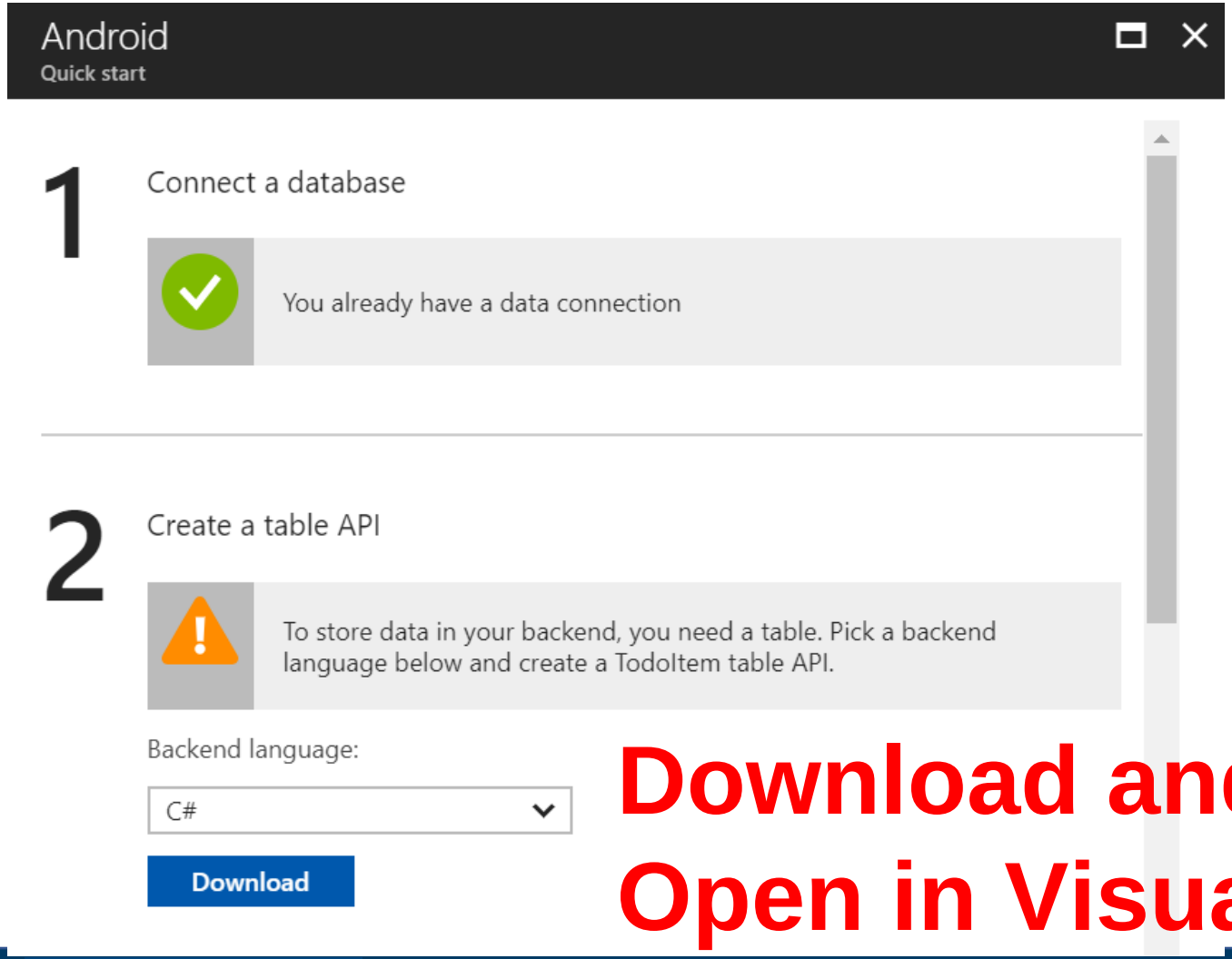


App Services BackEnd



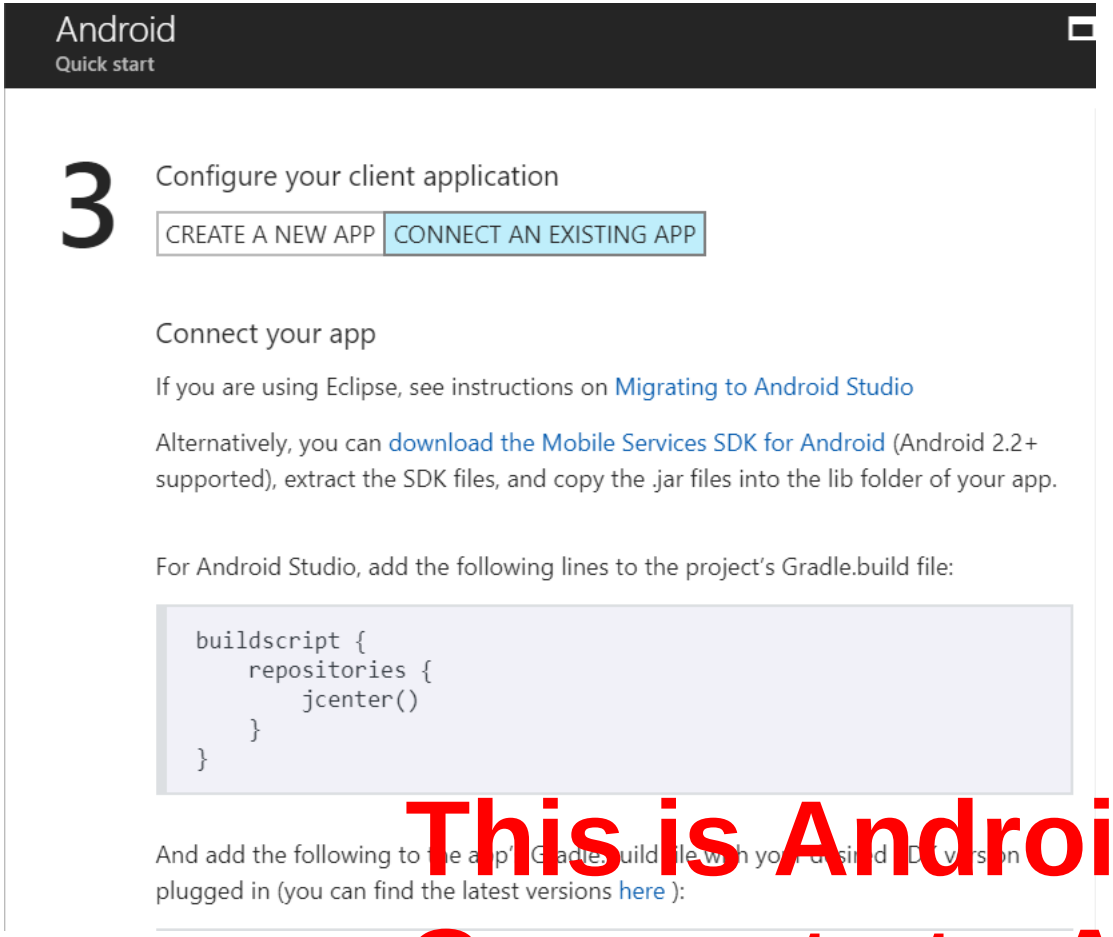
C# or Node.js
(This slides use C#)

App Services BackEnd



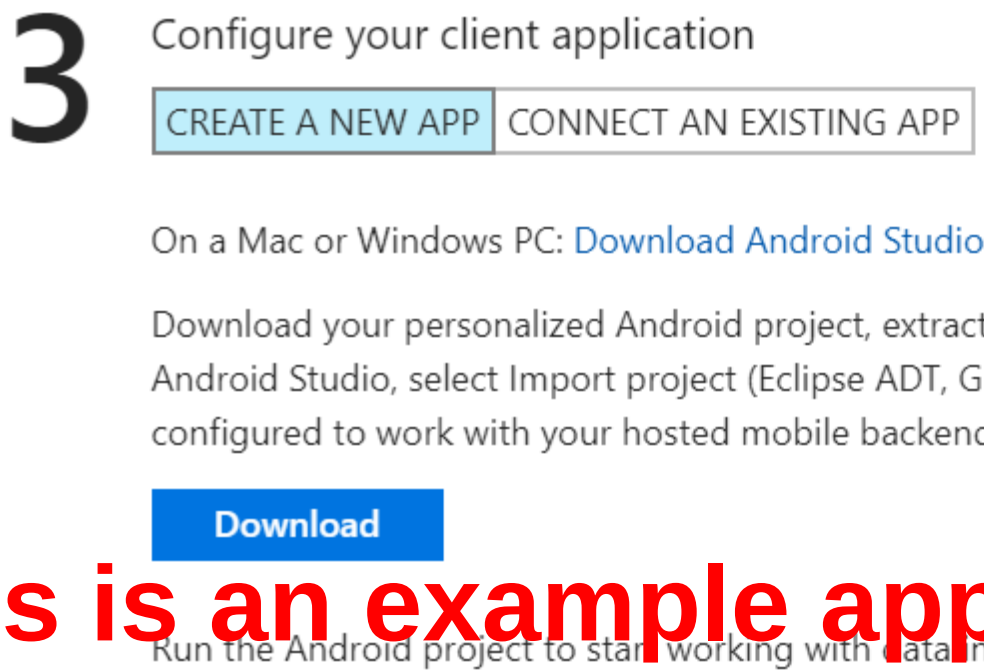
Download and Open in Visual Studio

App Services BackEnd



This is Android Connects to Azure

App Services BackEnd

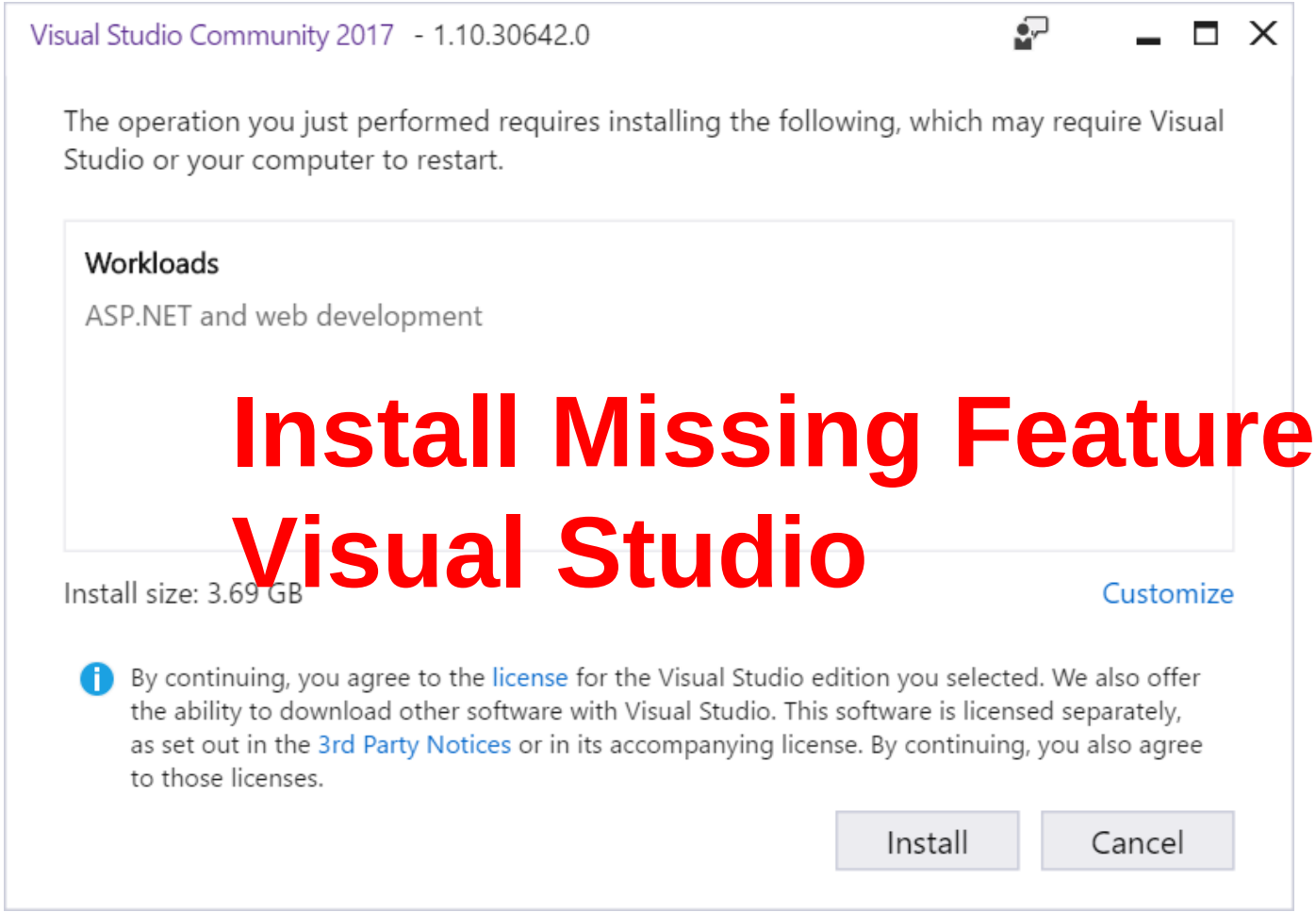


This is an example app of Android Connecting to Azure. Better Download it to understand the API.

Then, open the download solution file (Not Android example app) using Visual Studio
(Log in with your account, same as Azure)

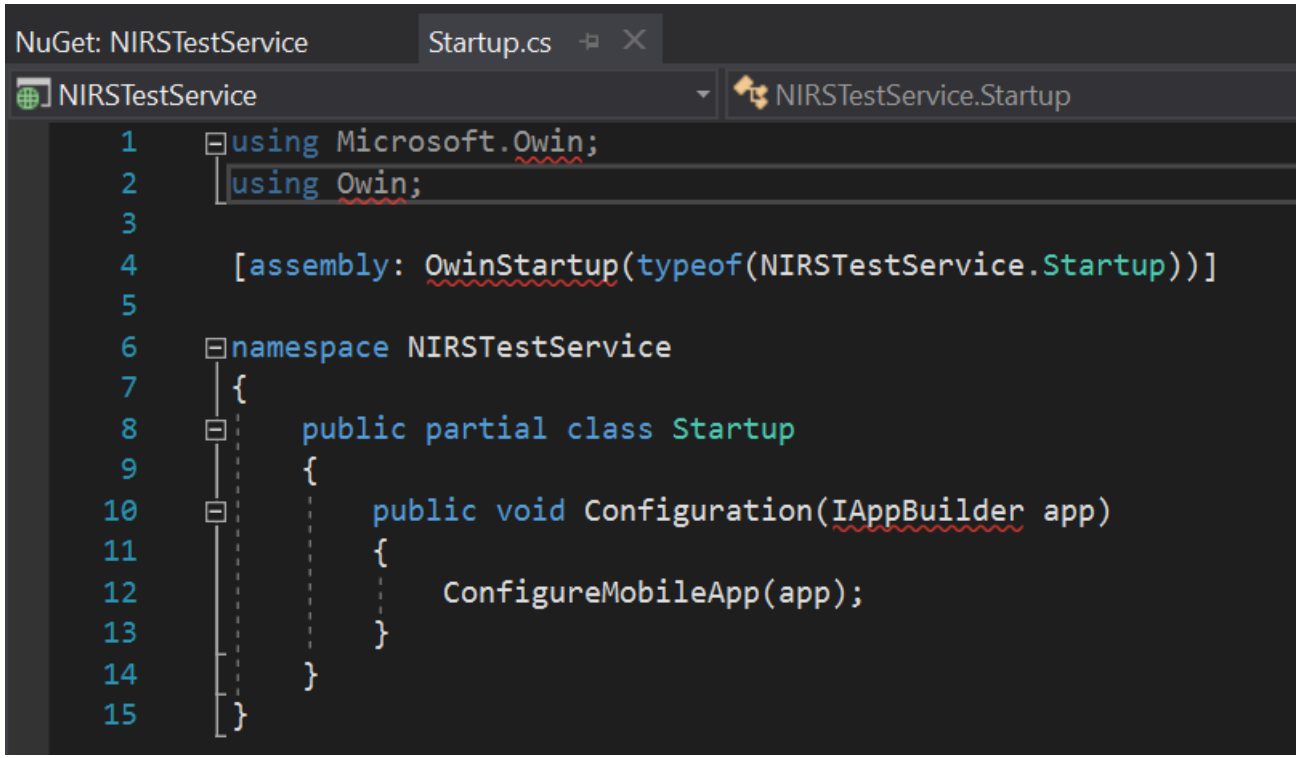
Note: Visual Studio 2017 v15.2 has bugs. So update it to 15.3

App Services BackEnd



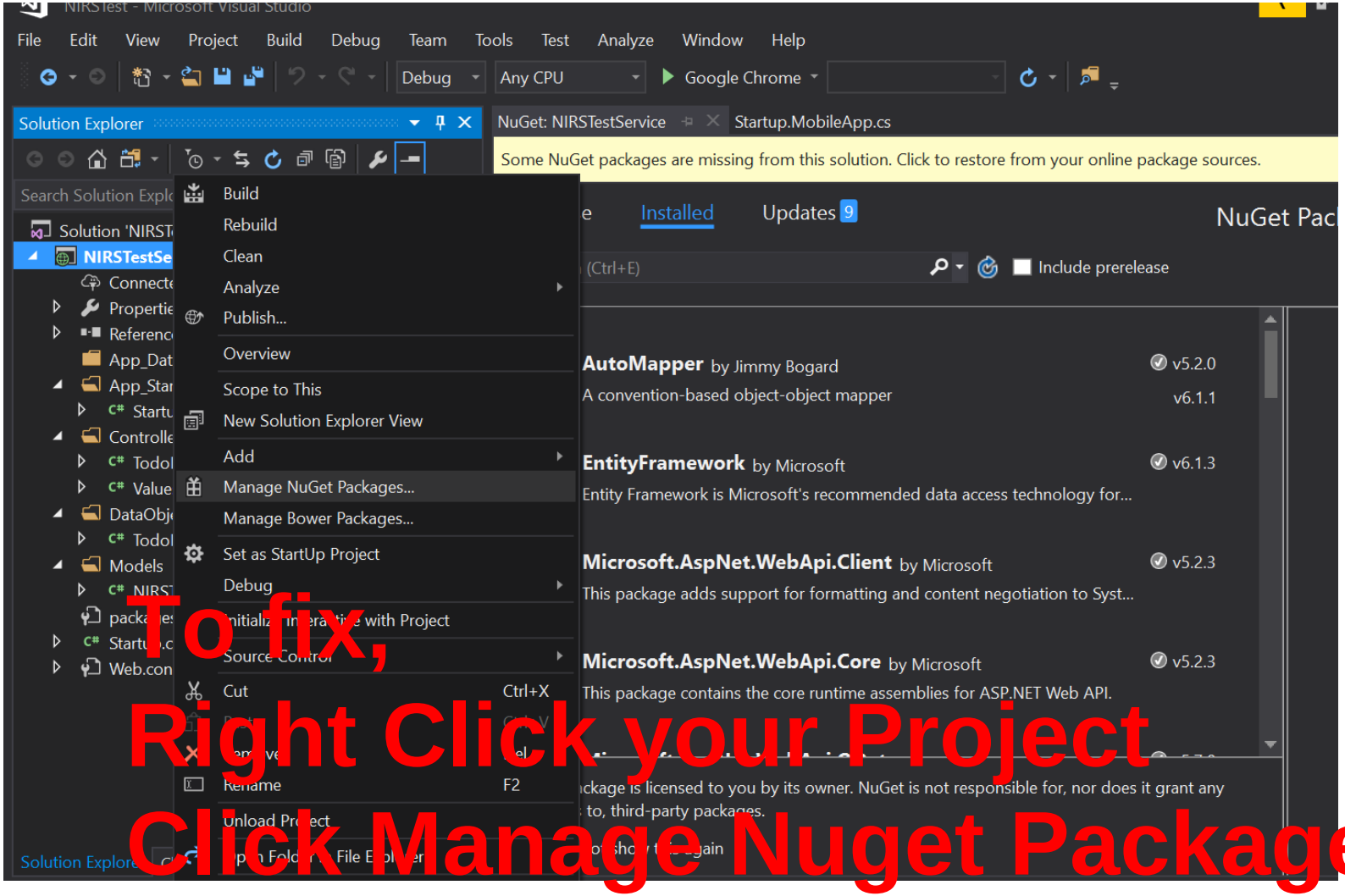
Install Missing Features in Visual Studio

Install SDK



At first, many items may be missing.
Red underline.

Install SDK



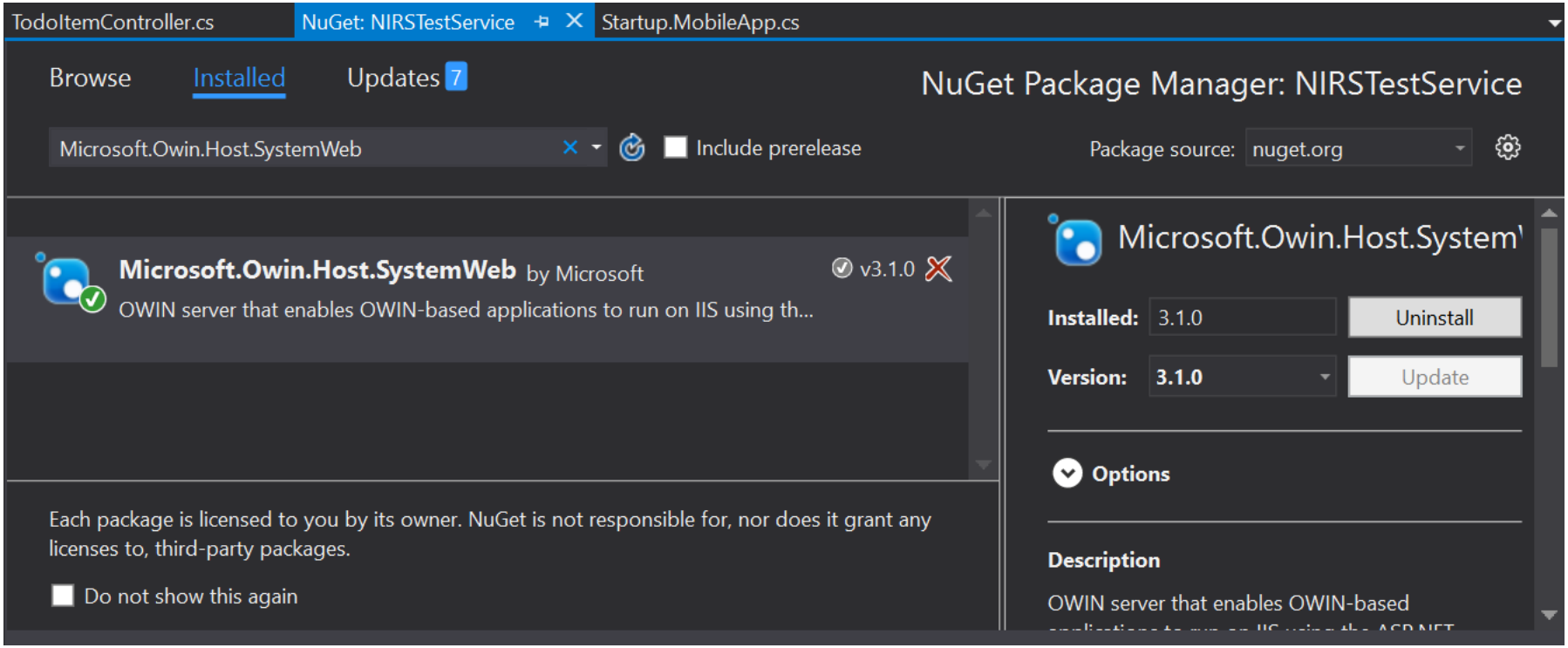
To fix,
Right Click your Project
Click Manage Nuget Packages

Install SDK

Install whatever you need.

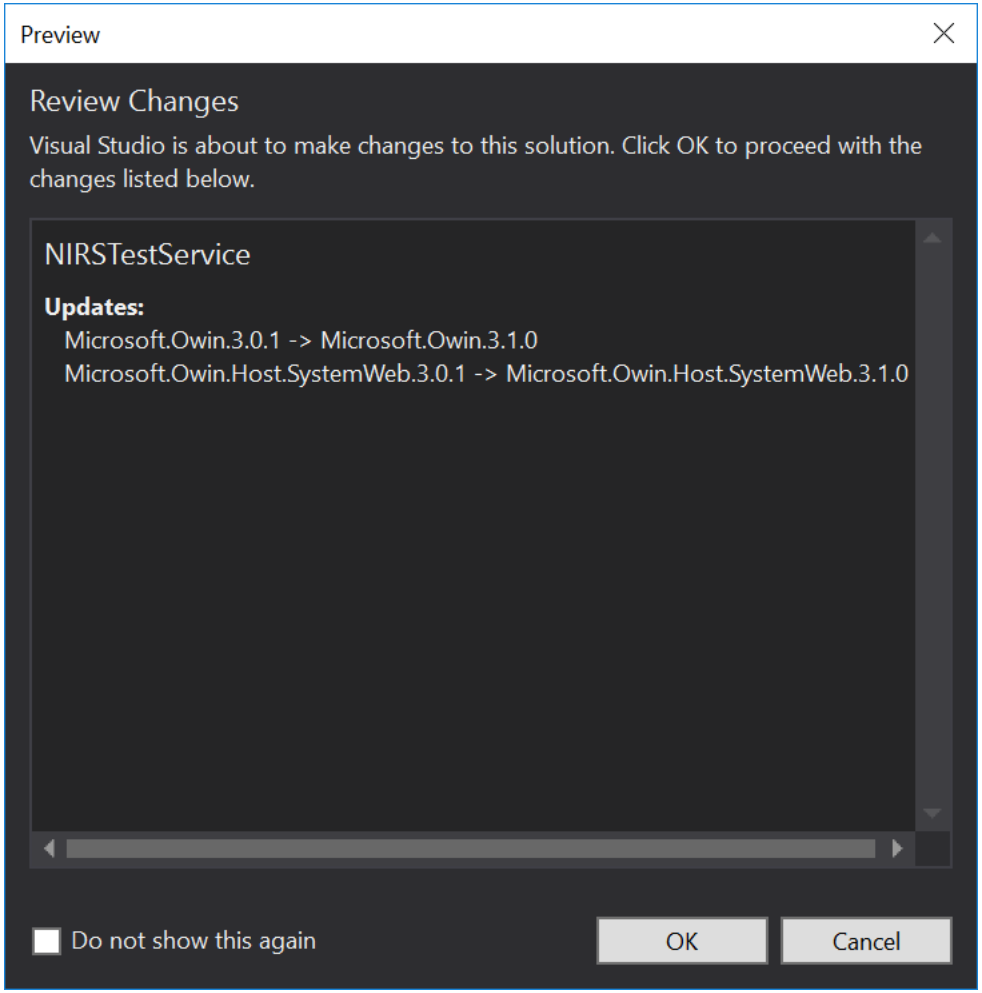
Depends on your Visual Studio State

Install SDK



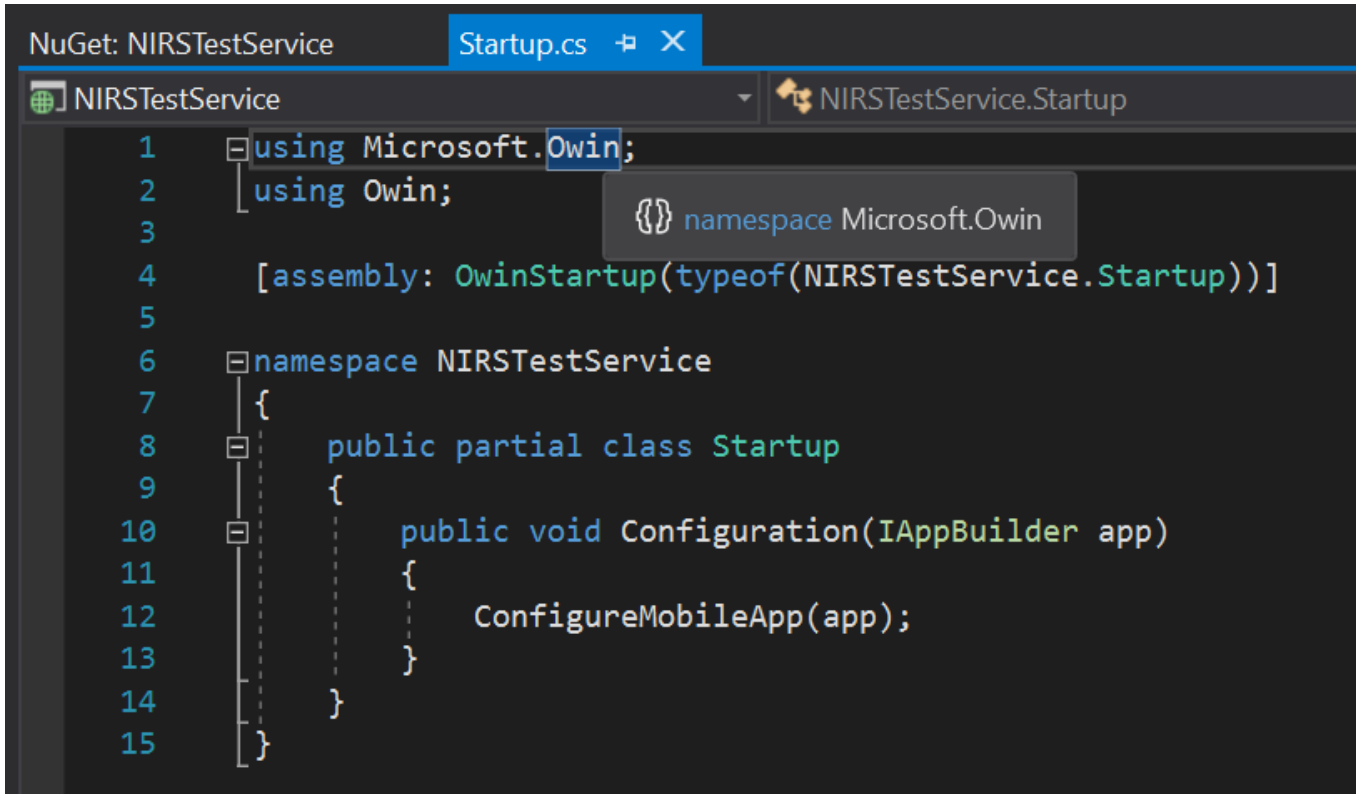
Probably need to update this to fix errors (in Chu's case)

Install SDK



Apply changes

Install SDK



Then everything should work

Initialise BackEnd

Make sure all files are identified well by VS (No red underlined code)

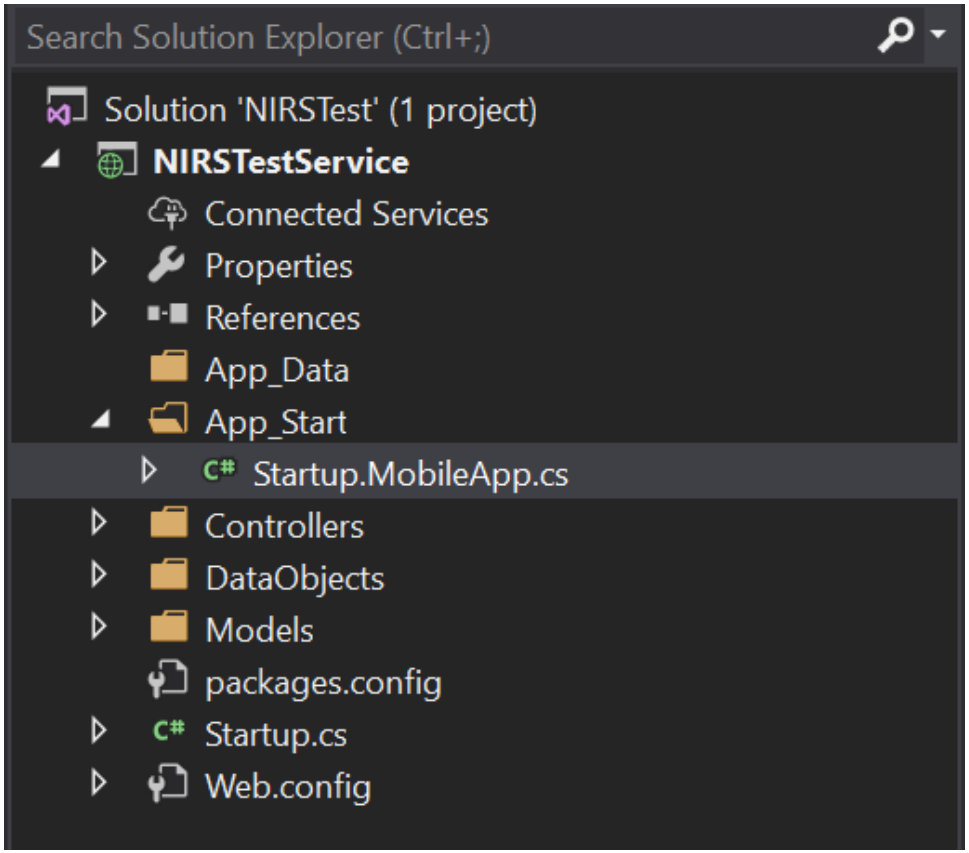
Initialise BackEnd

Initialize the server project

A .NET backend server project is initialized similar to other ASP.NET projects, by including an OWIN startup class. Ensure that you have referenced the NuGet package `Microsoft.Owin.Host.SystemWeb`. To add this class in Visual Studio, right-click on your server project and select **Add** > **New Item**, then **Web** > **General** > **OWIN Startup class**. A class is generated with the following attribute:

DON’T do this. Since you already have one.

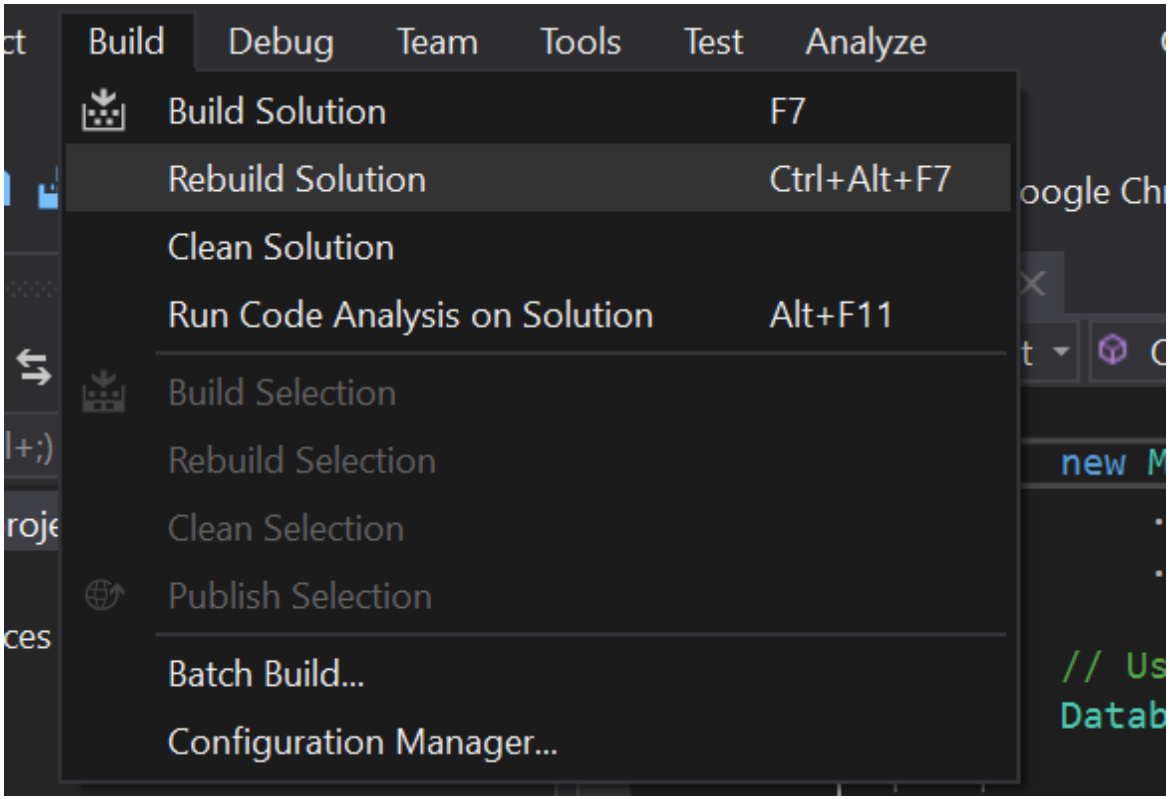
Initialise BackEnd



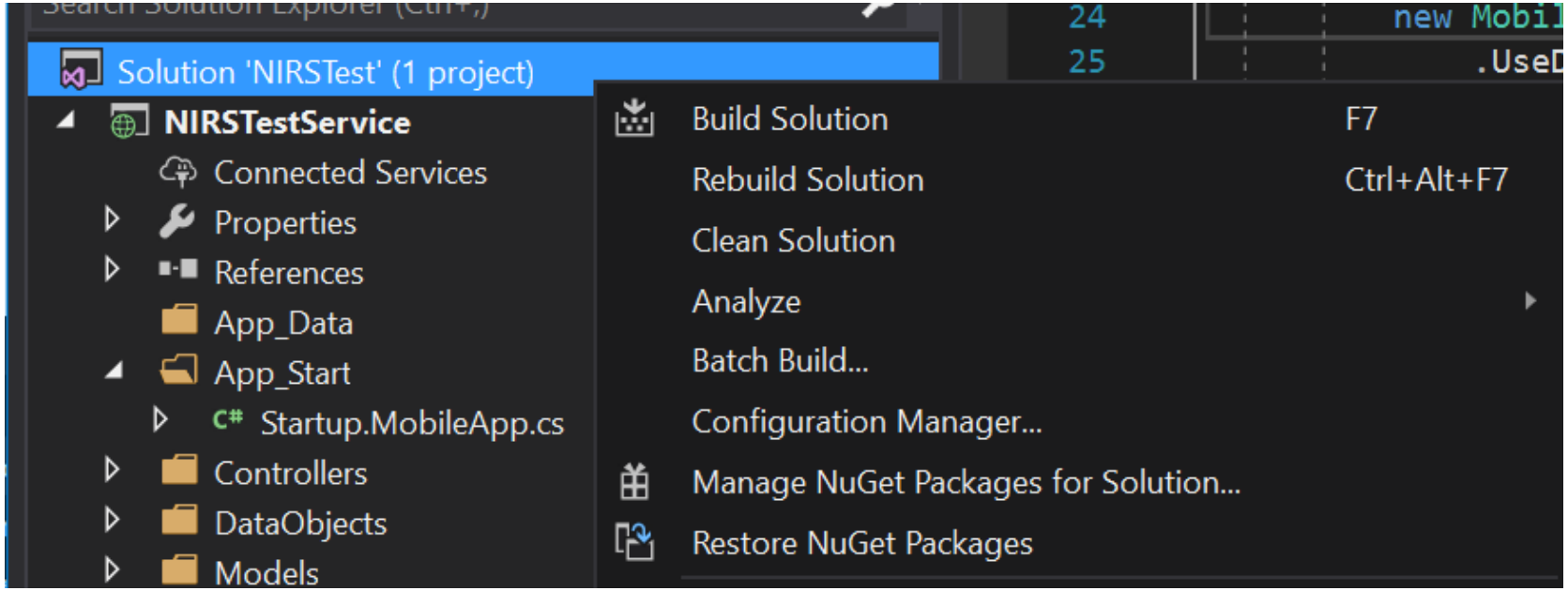
You already have it here.

How to publish your BackEnd online?

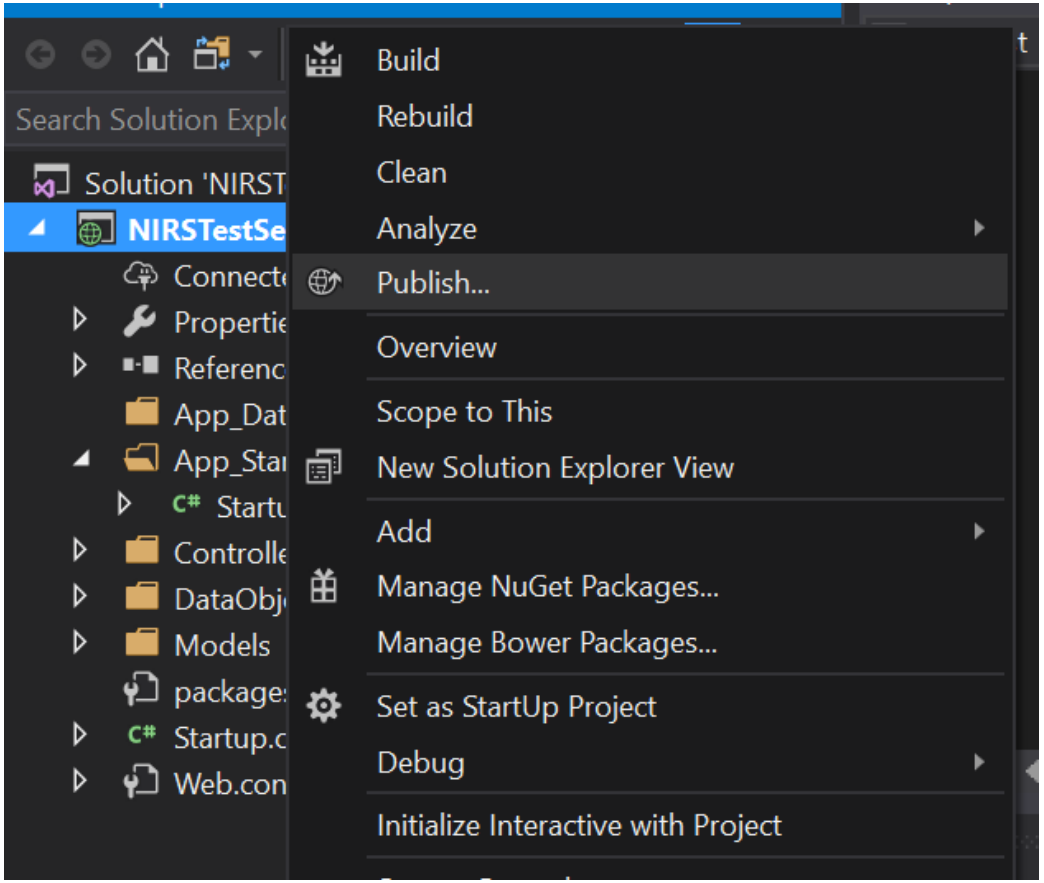
1. Rebuild


Build
Debug
Team
Tools
Test
Analyze
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Run Code Analysis on Solution Alt+F11
Build Selection
Rebuild Selection
Clean Selection
Publish Selection
Batch Build...
Configuration Manager...

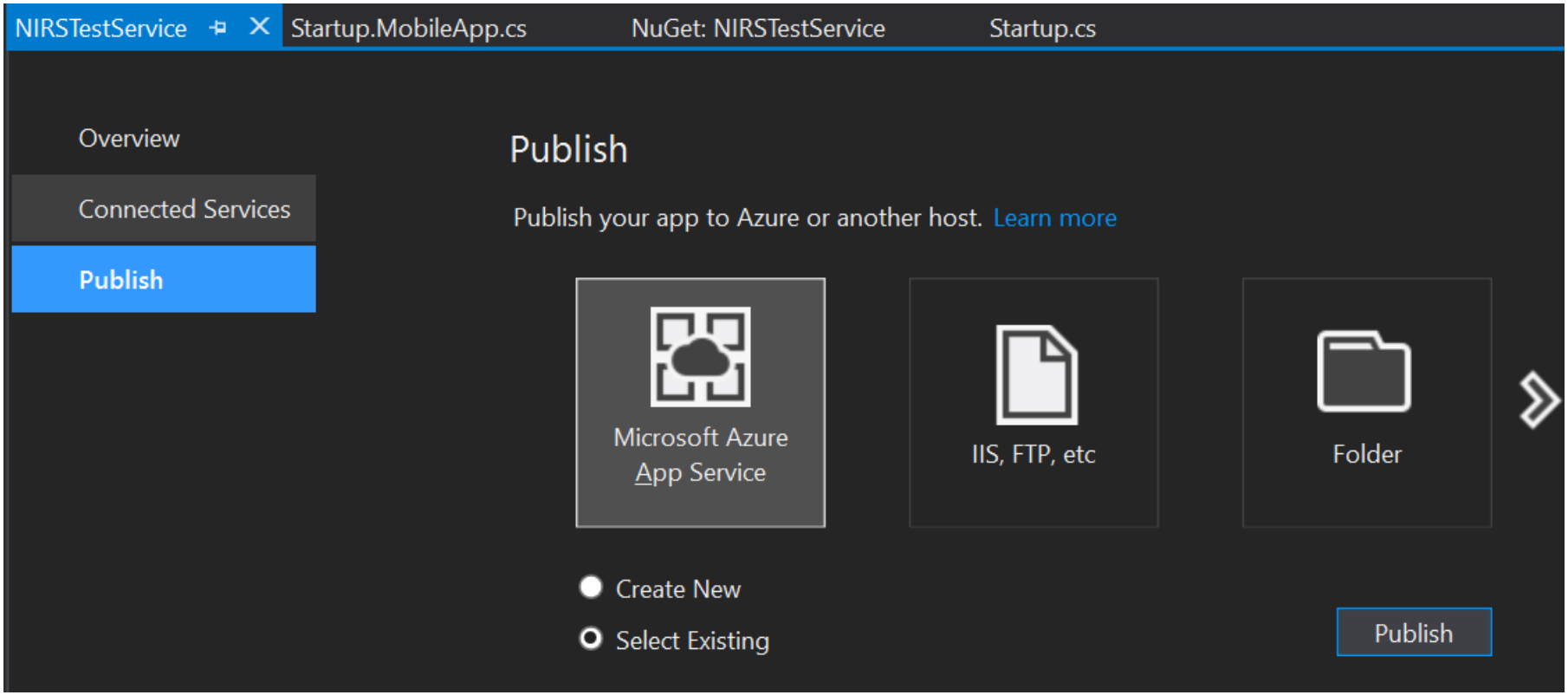
2. Restore NuGet



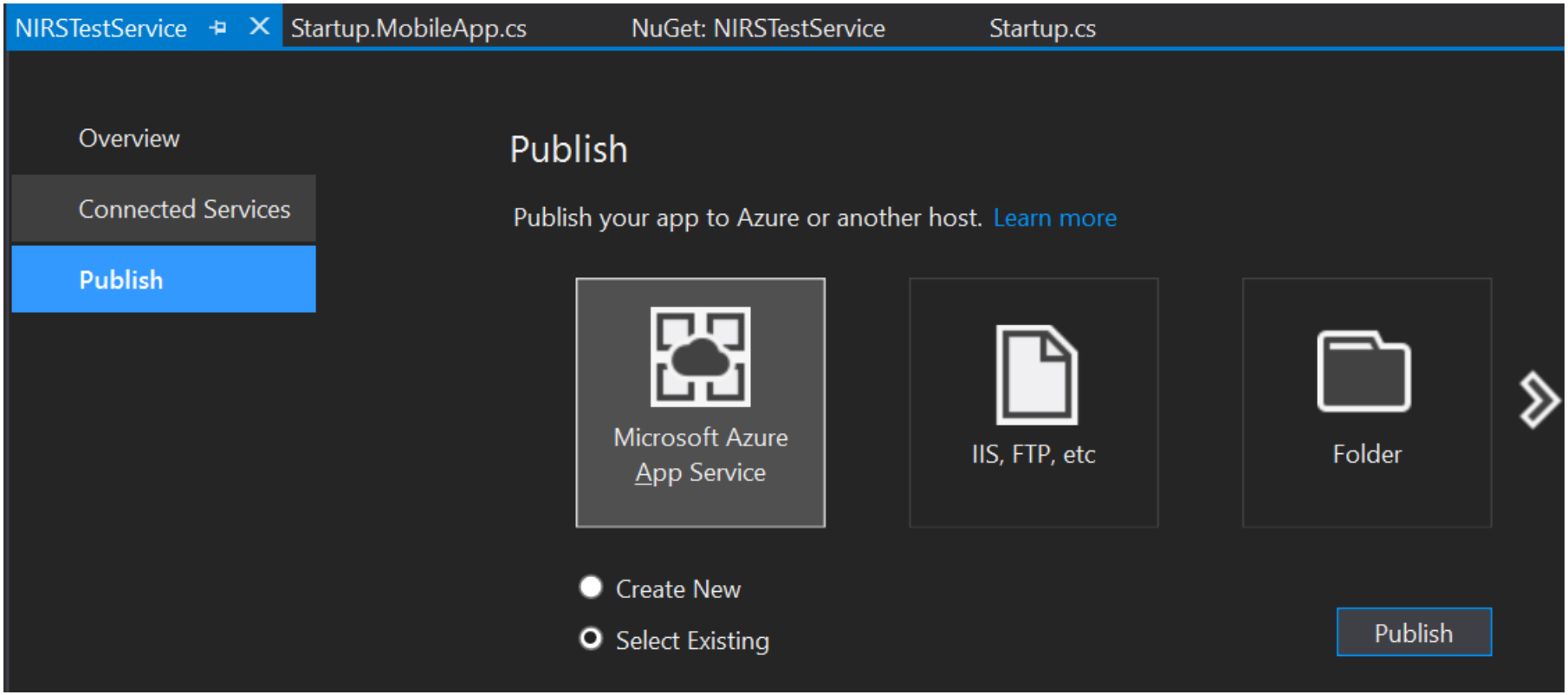
3. Publish


Search Solution Explo
Solution 'NIRST
NIRSTestSe
Connecte
Propertie
Referenc
App_Dat
App_Star
Startu
Controlle
DataObj
Models
package
Startup.c
Web.con
Build
Rebuild
Clean
Analyze
Publish...
Overview
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Manage Bower Packages...
Set as StartUp Project
Debug
Initialize Interactive with Project

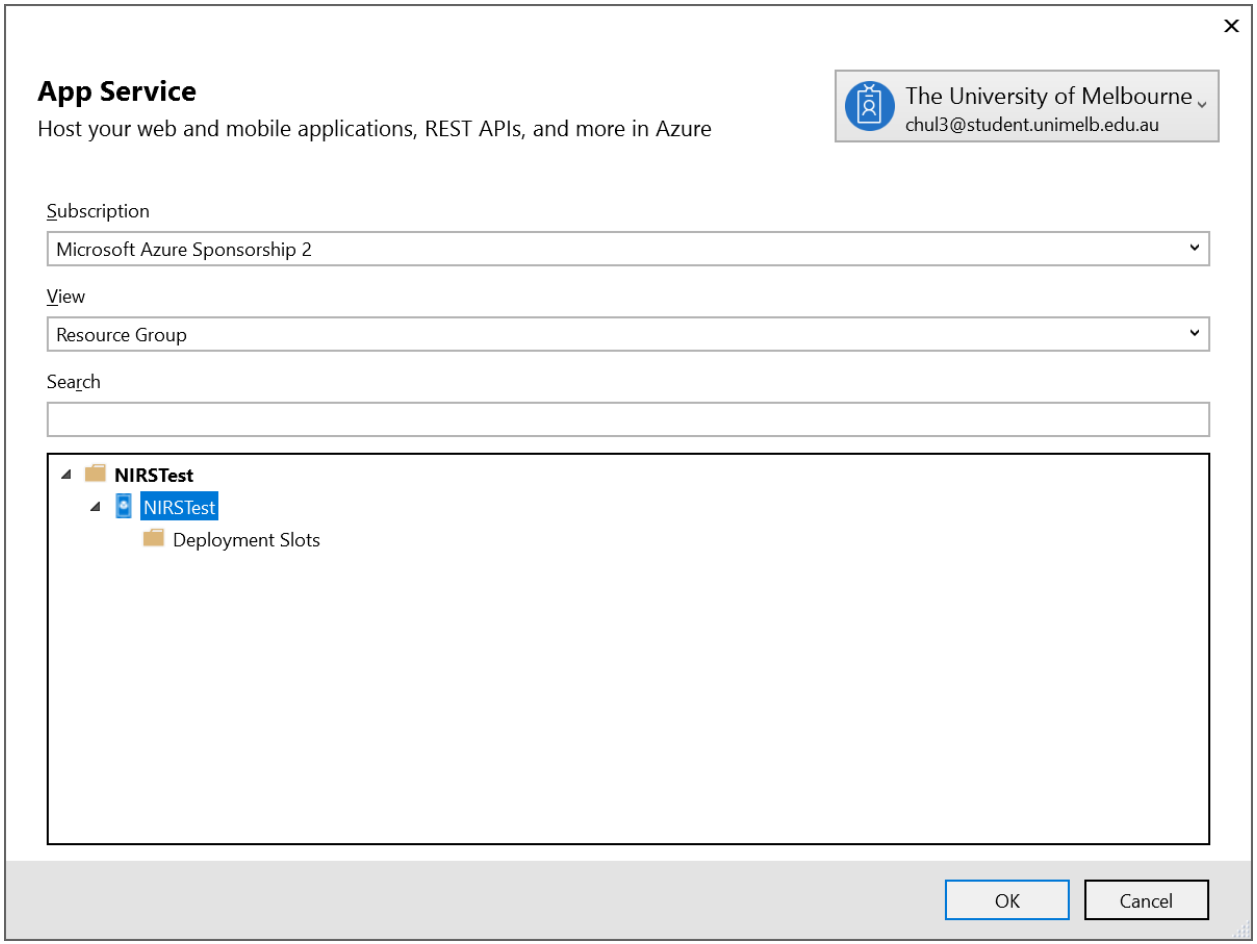
4. Select Existing App Service



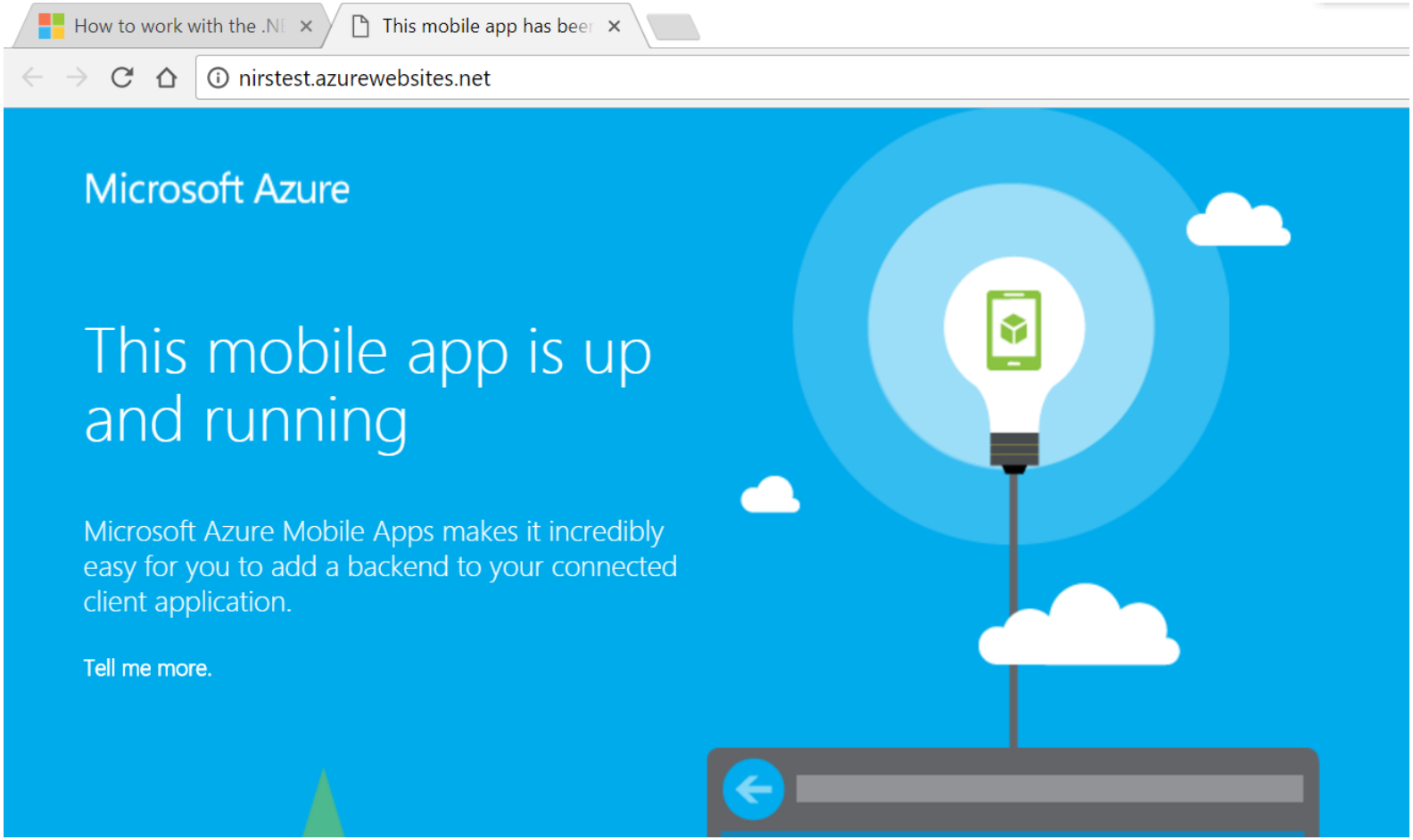
4. Select Existing App Service



5. Find your Azure App Service



6. Then you will see



To access online data on Azure:

You must implement a

Table Controller.

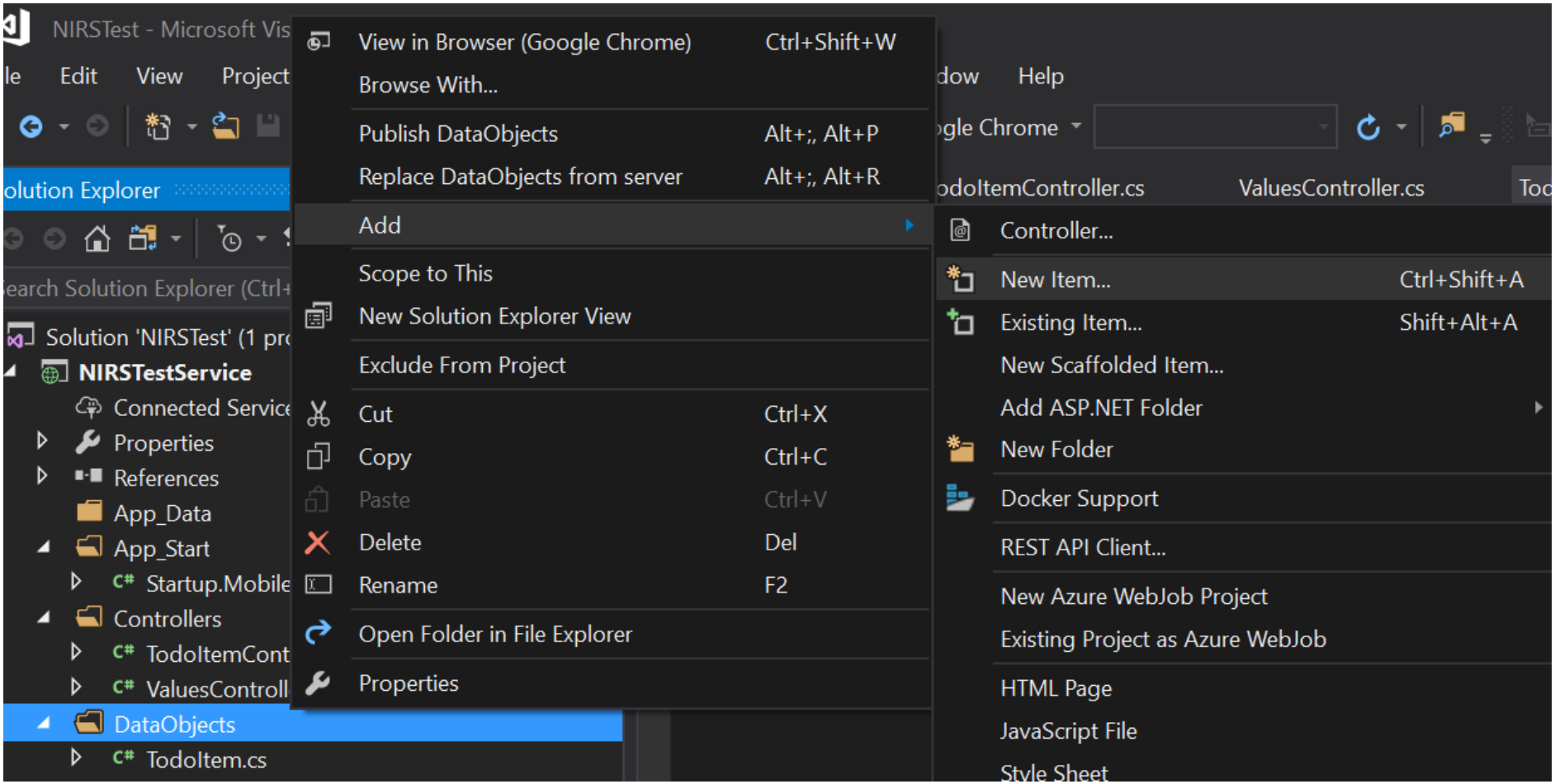
To use a Table Controller, we need to create:

a Data Transfer Object (DTO)
a Table Controller
a Code-First Migration

Data Transfer Object

This is the object design same as your database schema

Data Transfer Object



If you want to add, use C# class

Data Transfer Object

TodoItem.cs | NIRSTestService | Web.config | TodoItemController.c

NIRSTestService | NIRSTestService.DataObjects

```
1   using Microsoft.Azure.Mobile.Server;
2
3   namespace NIRSTestService.DataObjects
4   {
5       public class TodoItem : EntityData
6       {
7           public string Text { get; set; }
8           //public int IntField { get; set; }
9           // public double DoubleField { get; set; }
10
11          public bool Complete { get; set; }
12      }
13  }
```

TodoItem is an existing example

Data Transfer Object

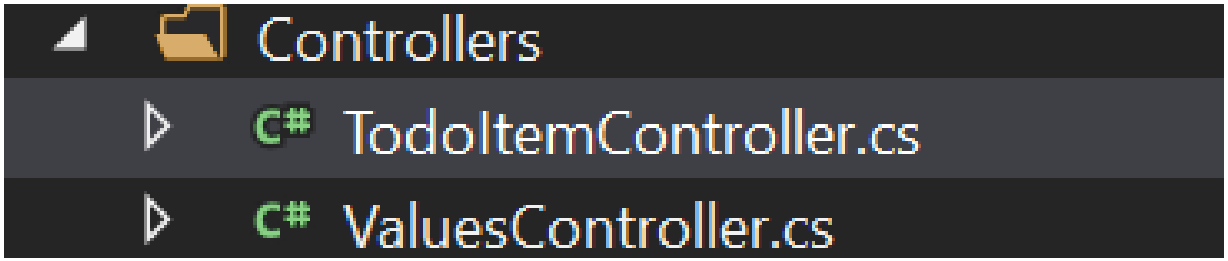
```
public class NIRSTestInitializer : CreateDatabaseIfNotExists<NIRSTestContext>
{
    protected override void Seed(NIRSTestContext context)
    {
        List<TodoItem> todoItems = new List<TodoItem>
        {
            new TodoItem { Id = Guid.NewGuid().ToString(), Text = "First item", Complete = false },
            new TodoItem { Id = Guid.NewGuid().ToString(), Text = "Second item", Complete = false },
        };
```

Startup.MobileApp.cs may do something about the DTO

To use a Table Controller, we need to create:

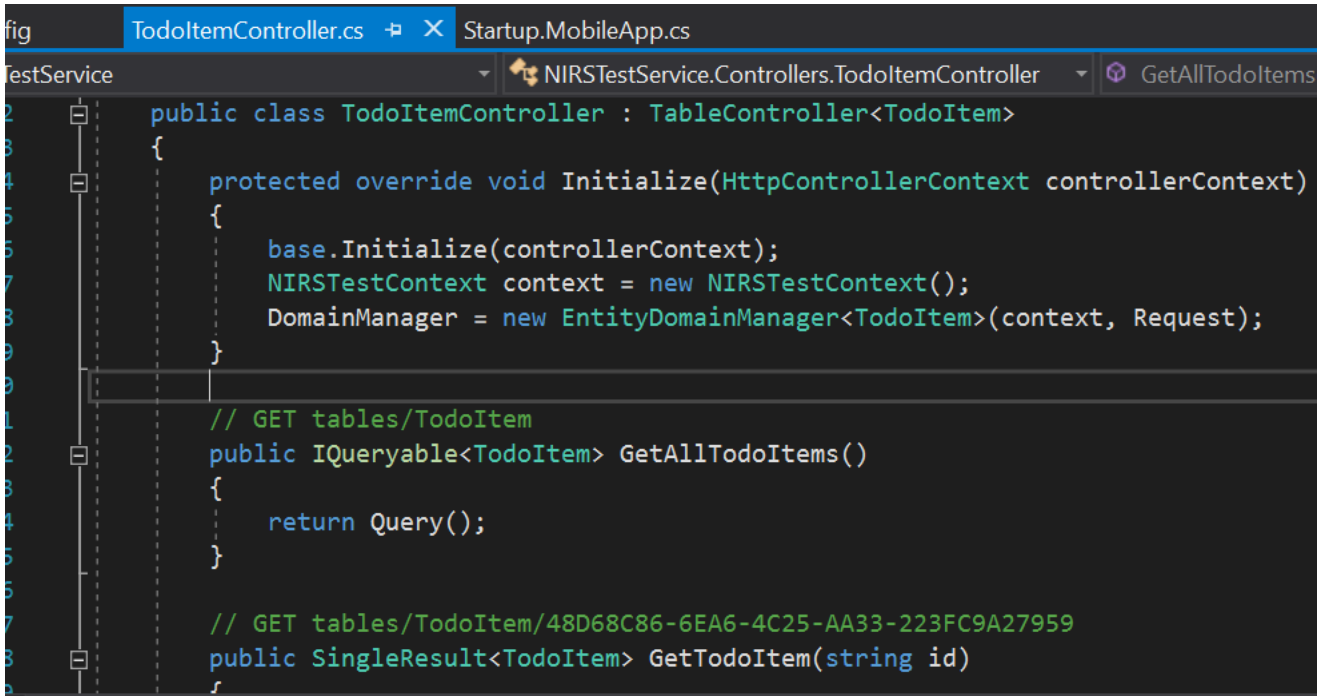
- ✓ a Data Transfer Object (DTO)
- **a Table Controller**
- **a Code-First Migration**

Table Controller



TodoItemController is an existing example

Table Controller



TodoItemController is an existing example

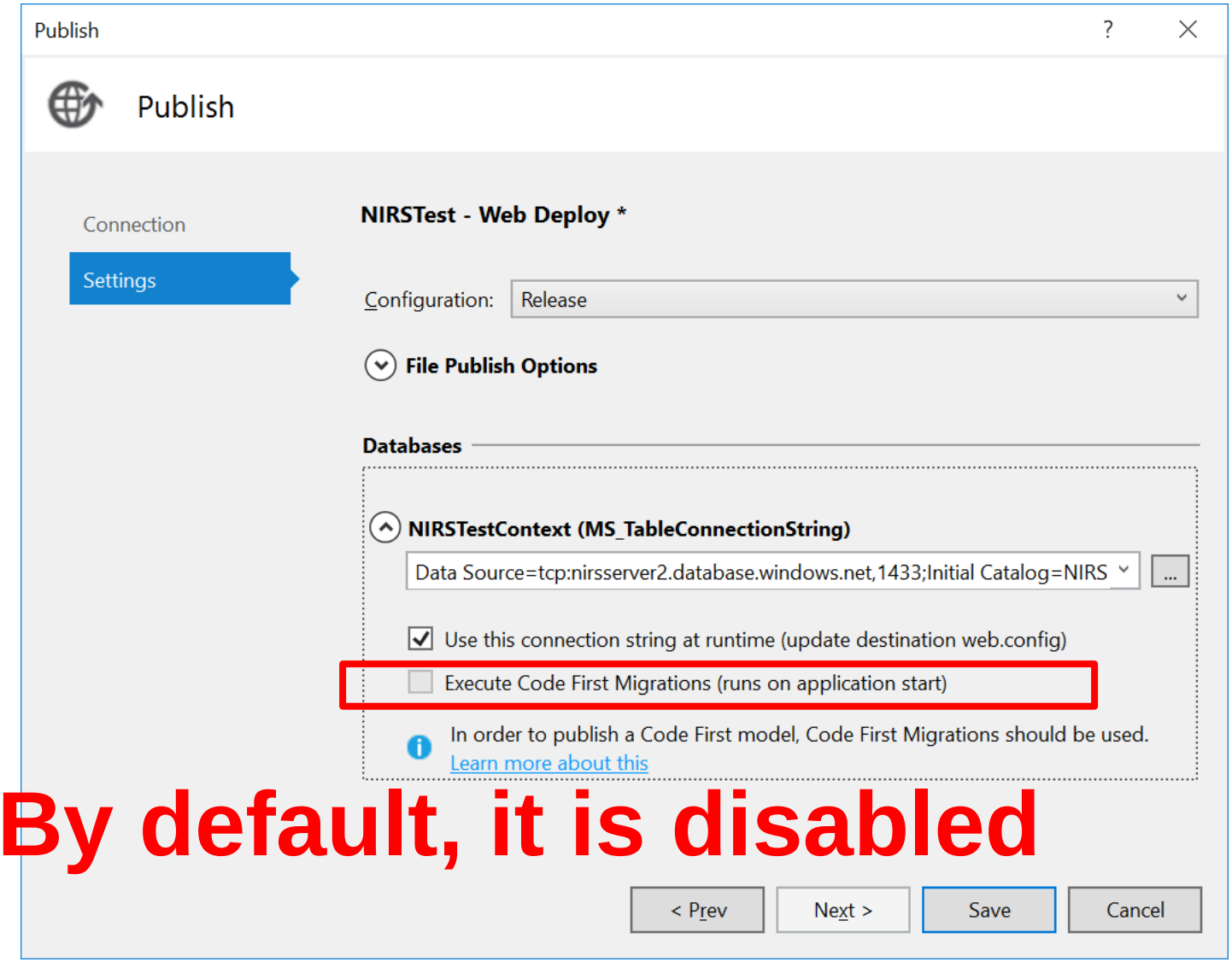
To use a Table Controller, we need to create:

- ✓ a Data Transfer Object (DTO)
- ✓ a Table Controller
- a Code-First Migration

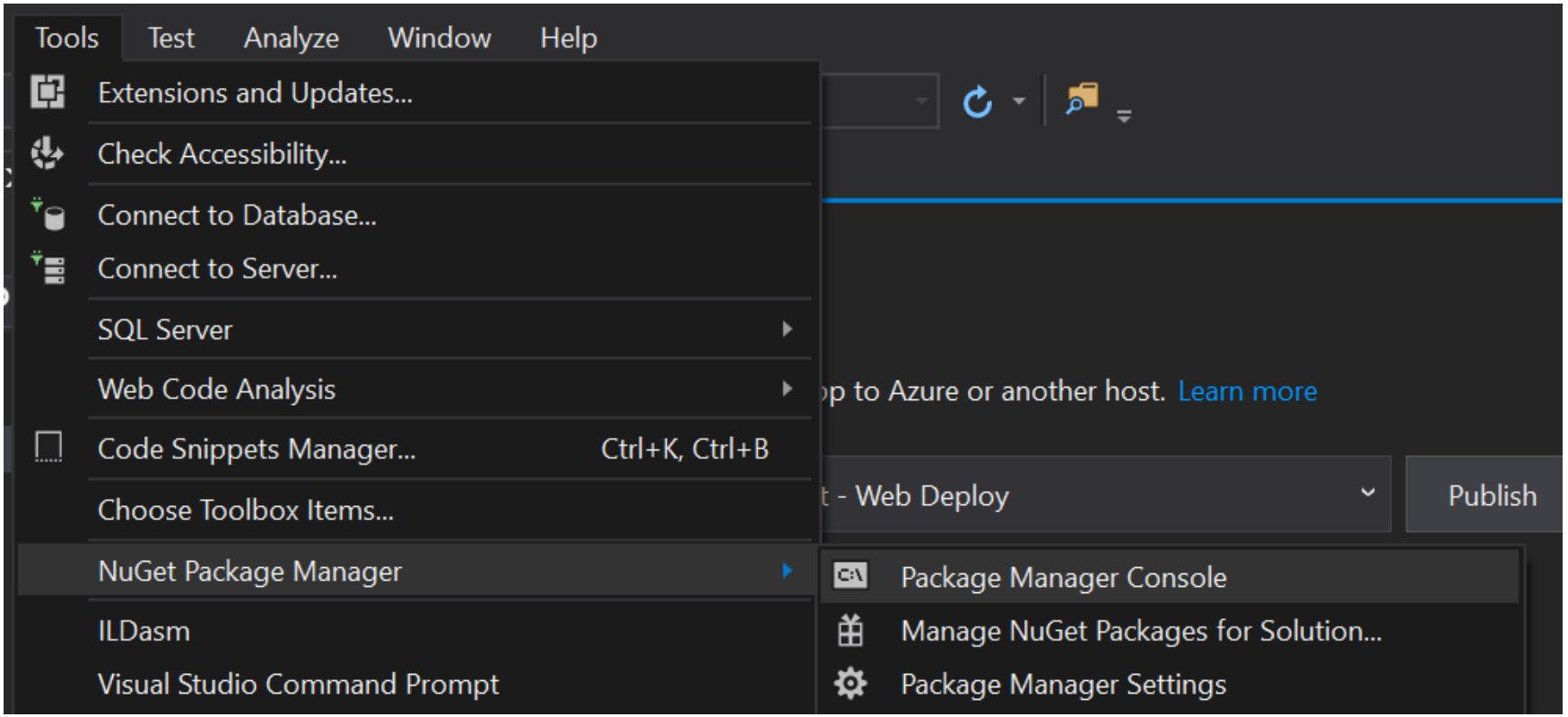
Code-First Migration

https://msdn.microsoft.com/en-us/library/dd394698#efcfmigrations

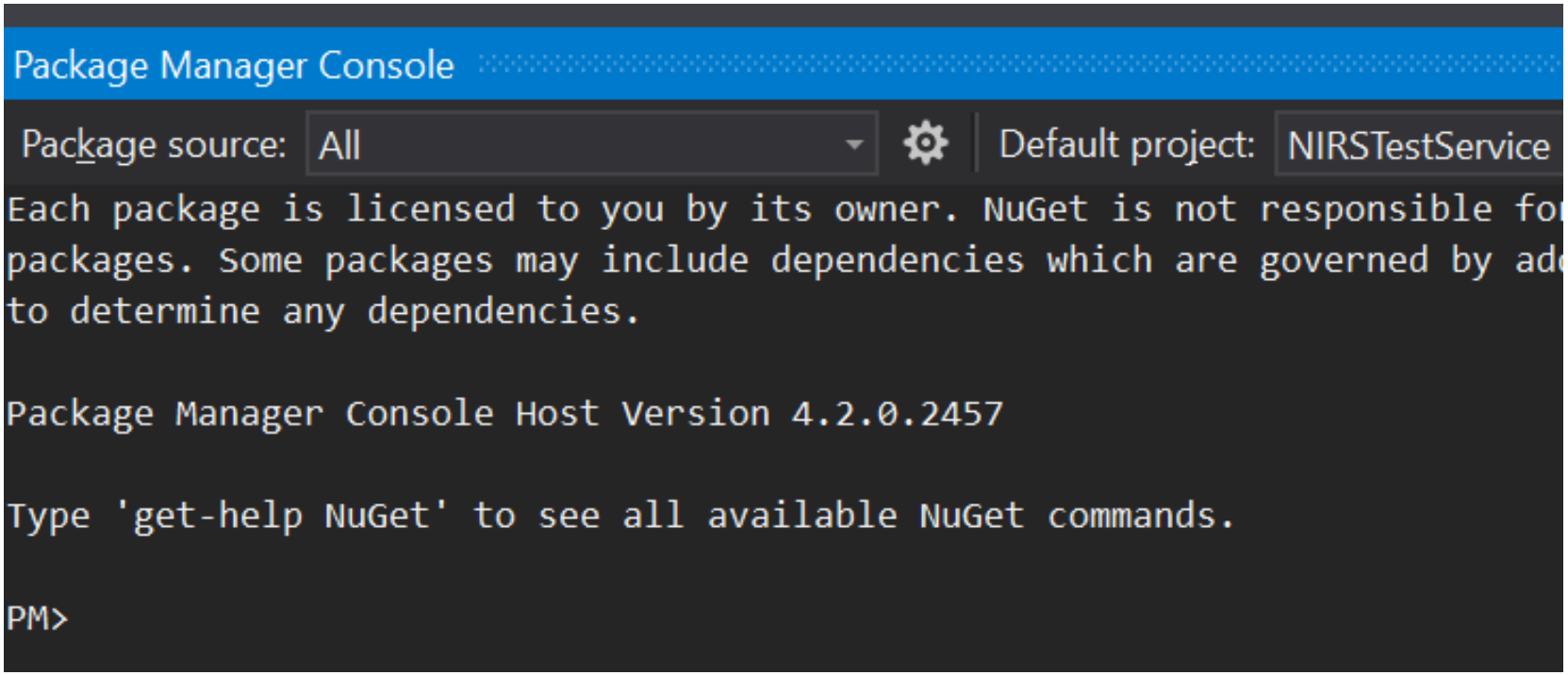
This is to make sure your online DB will upgrade to your latest design in Visual Studio



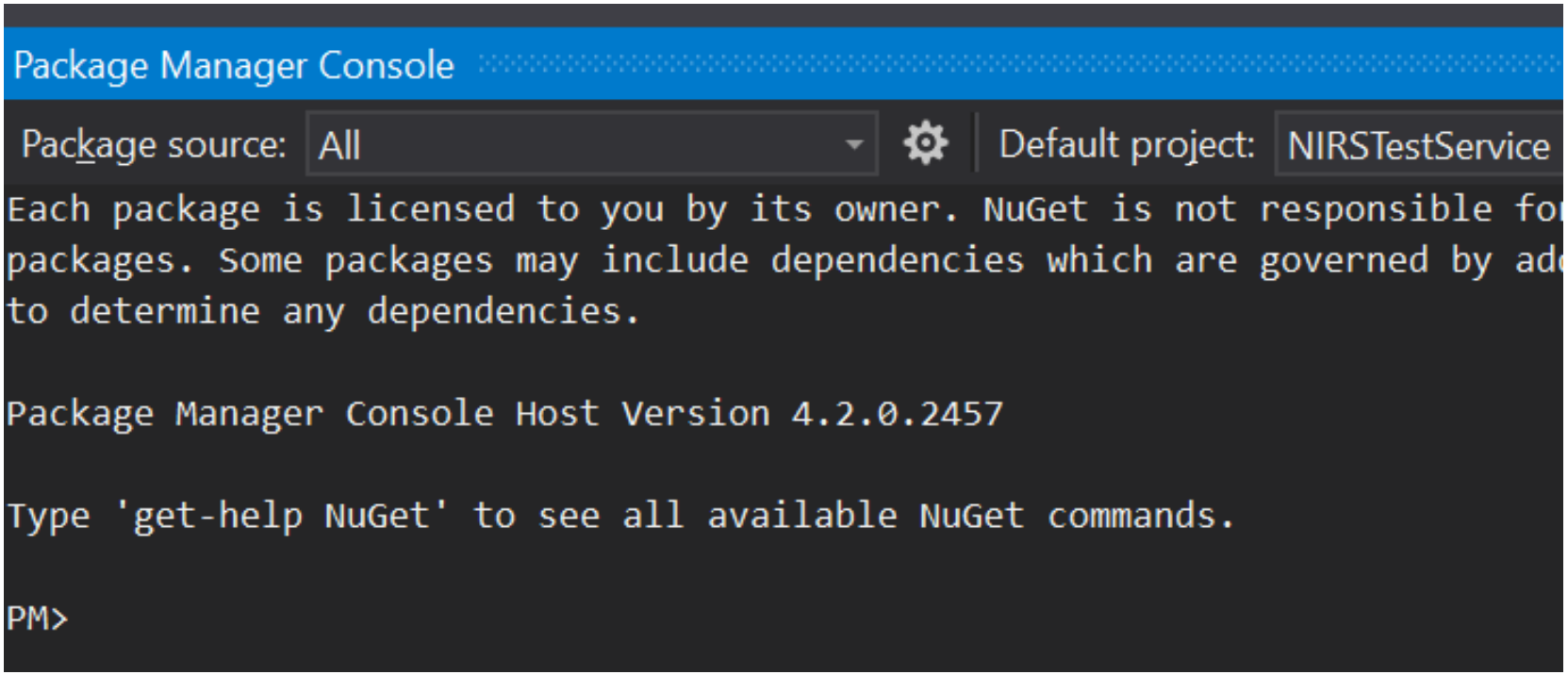
By default, it is disabled



Open this Console



You will see this



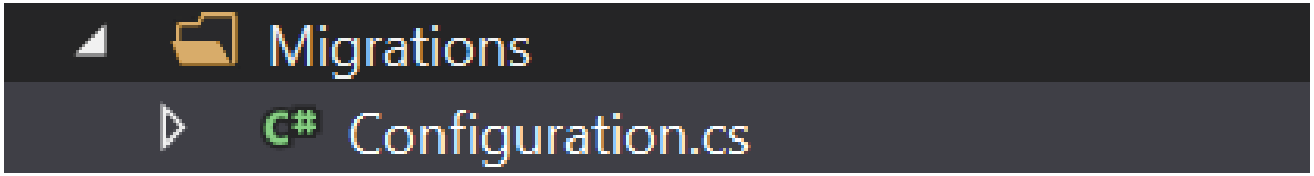
Enter: install-package entityframework

```
Package Manager Console Host Version 4.2.0.2457

Type 'get-help NuGet' to see all available NuGet commands.

PM> install-package entityframework
Package 'entityframework.6.1.3' already exists in project 'NIRSTestService'
Time Elapsed: 00:00:00.6106869
PM>
```

If already installed, you see this

```
PM> enable-migrations
Checking if the context targets an existing database...
Code First Migrations enabled for project NIRSTestService.
PM>
```

Input: enable-migrations



Then you see this

```
PM> add-migration
cmdlet Add-Migration at command pipeline position 1
Supply values for the following parameters:
Name: name1
Scaffolding migration 'name1'.
The Designer Code for this migration file includes a
 the changes to your model when you scaffold the next
include in this migration, then you can re-scaffold i
PM>
```

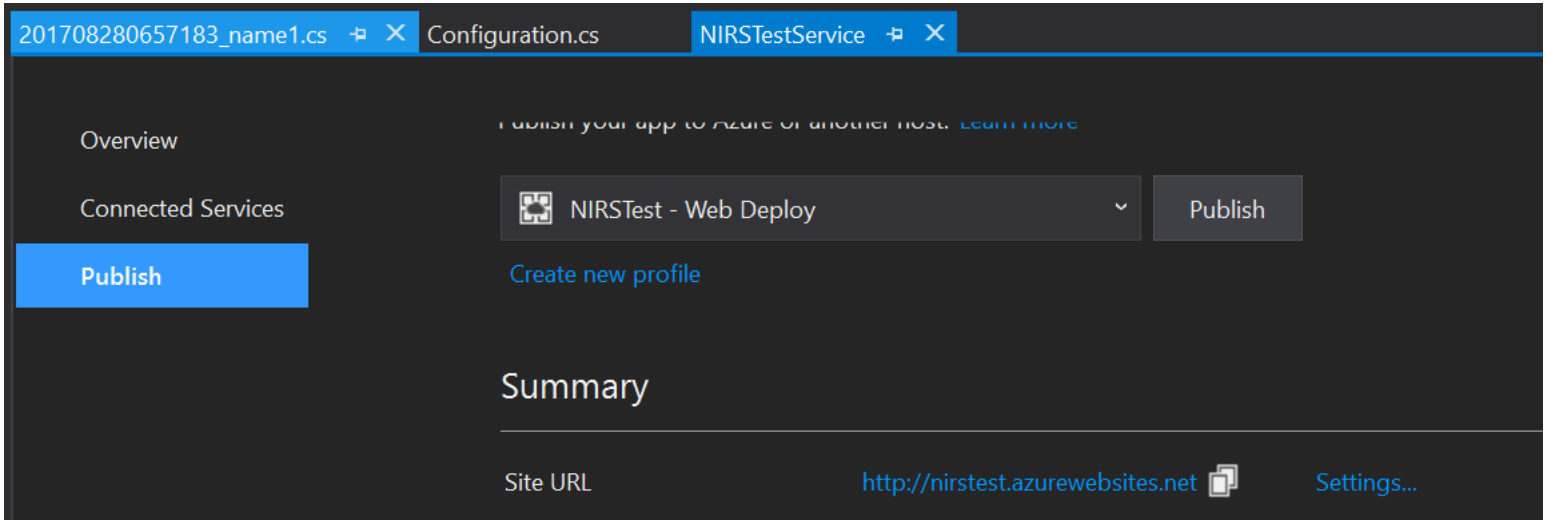
Then input: add-migration
Also put whatever name you like

Check: https://adrianhall.github.io/develop-mobile-apps-with-csharp-and-azure/chapter3/server/

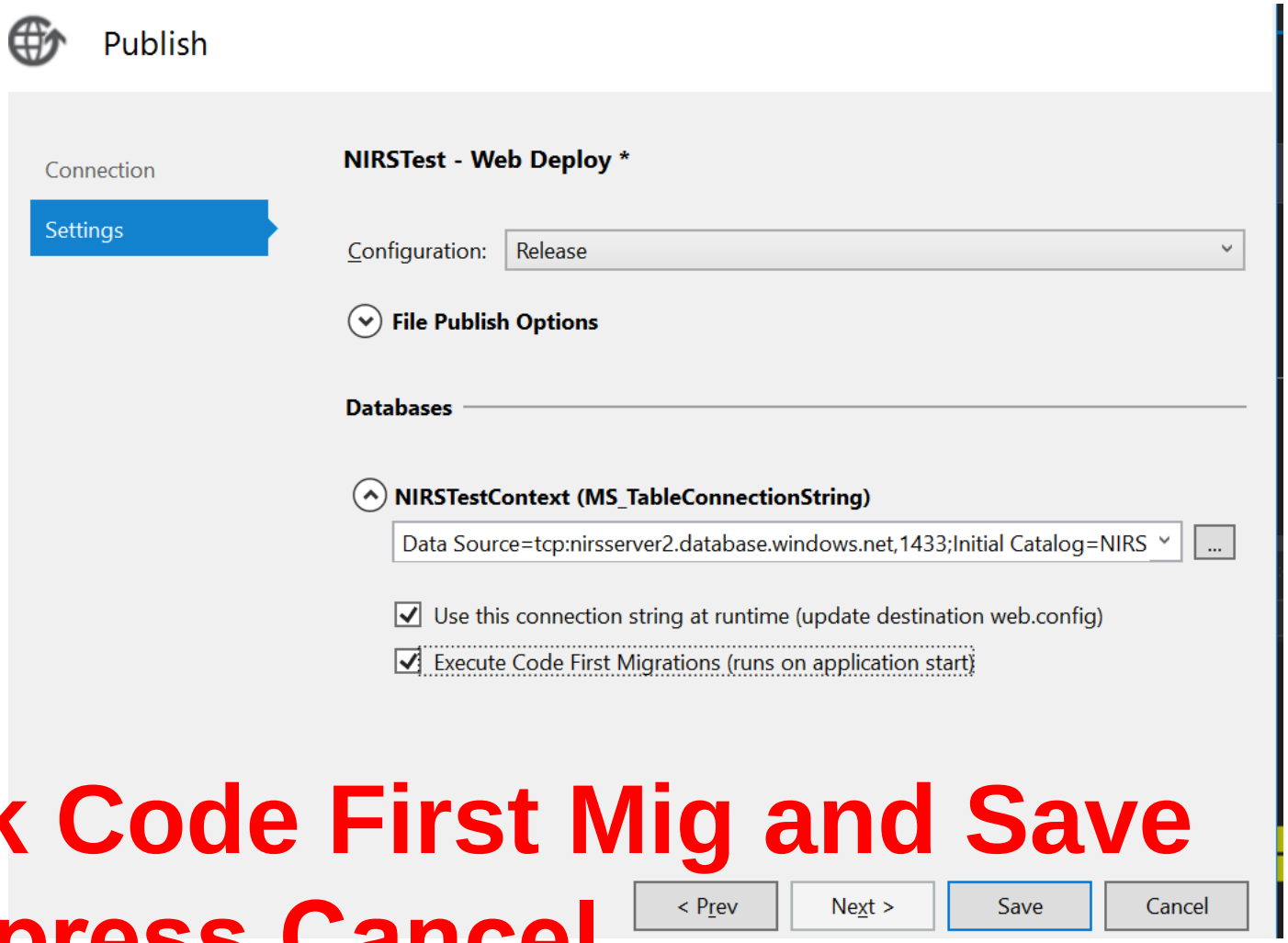
You should also use `update-database` to apply the change to the local database (if any):

```
C:\Users\adrian\Source\Repos\develop-mobile-apps-with-csharp-and-azure\Chapter3 [master ]> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201608140206437_AddExampleTable].
Applying explicit migration: 201608140206437_AddExampleTable.
Running Seed method.
C:\Users\adrian\Source\Repos\develop-mobile-apps-with-csharp-and-azure\Chapter3 [master ]>
```

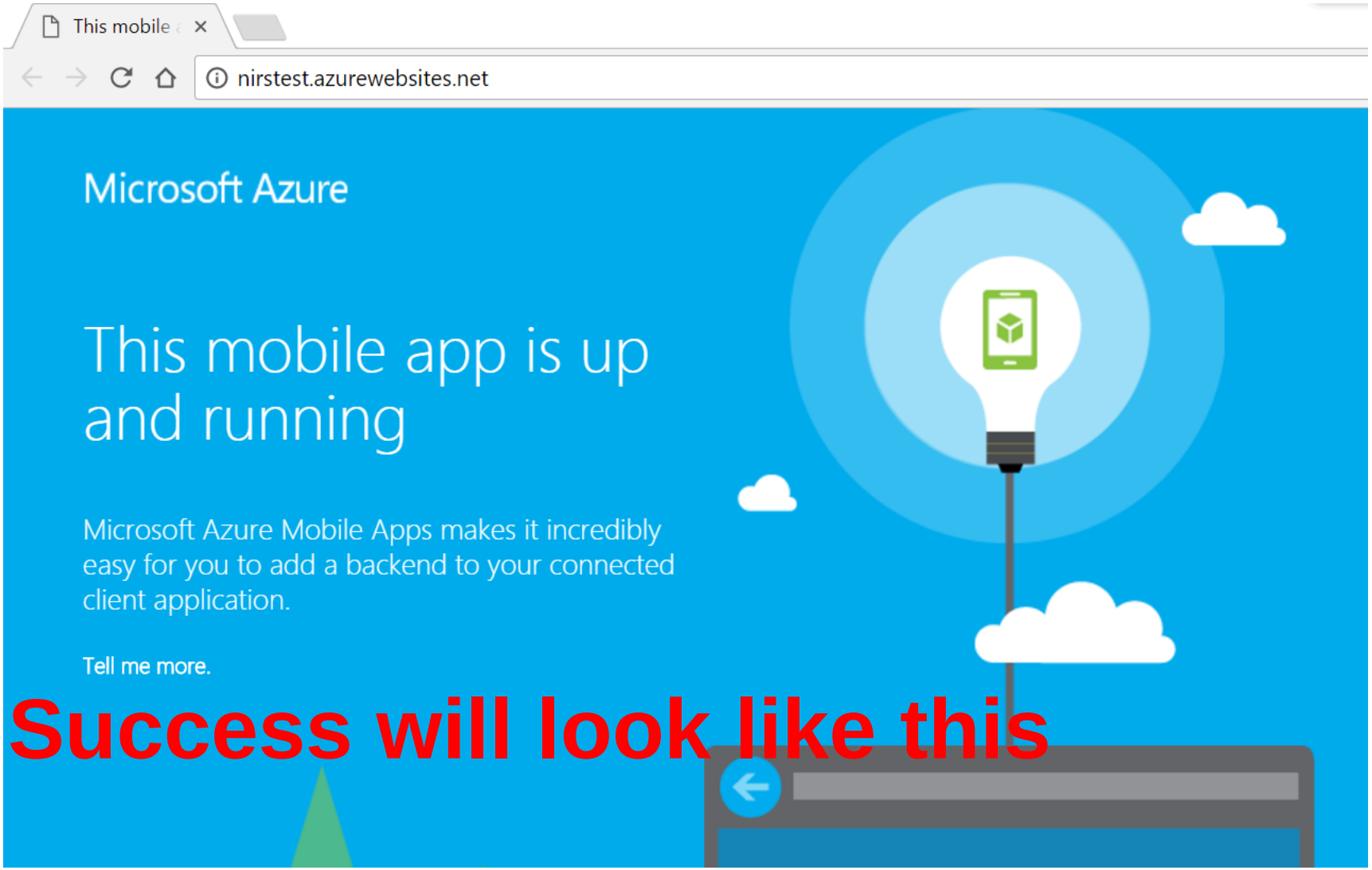
Well, Chu does not have a local one



Now go to Publish Settings



Tick Code First Mig and Save No press Cancel…



Success will look like this

Demo …

Exercise:

1. Design your app with Azure

2. Implement your BackEnd App Service

3. Let your BackEnd and FrontEnd communicate. C.f. Android Example app (page 12)

More learning directions:

1. Finish your assignment with sensors and Azure

See you next week

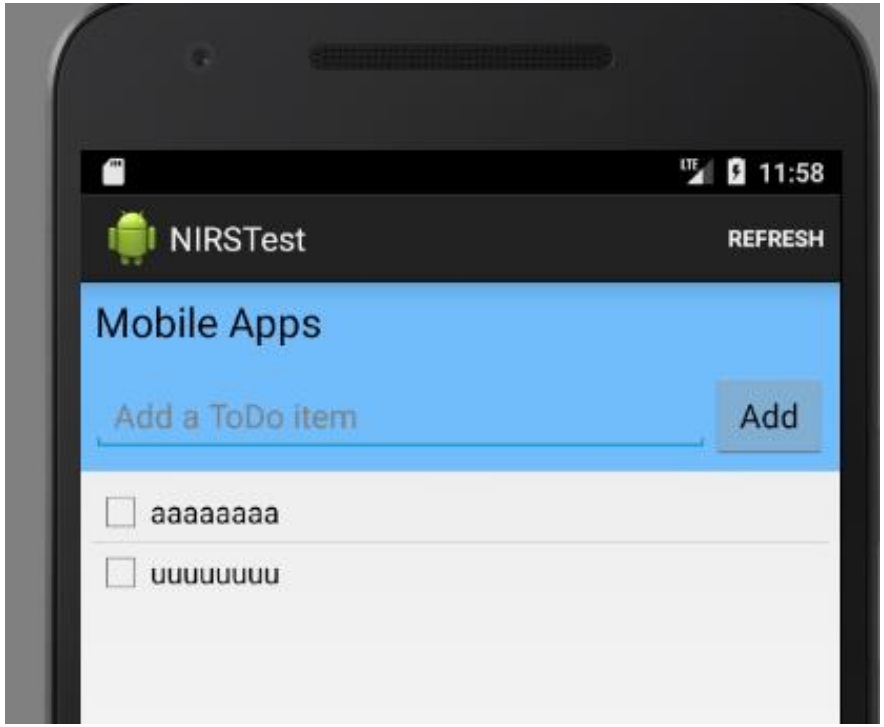
COMP 90018

Tutorial on Android Development

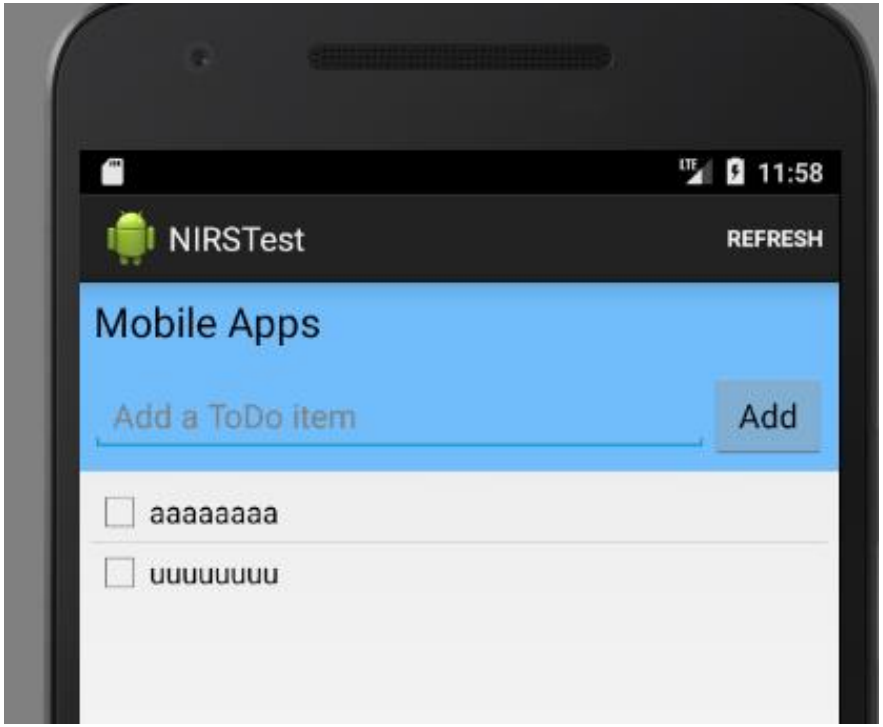
Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au

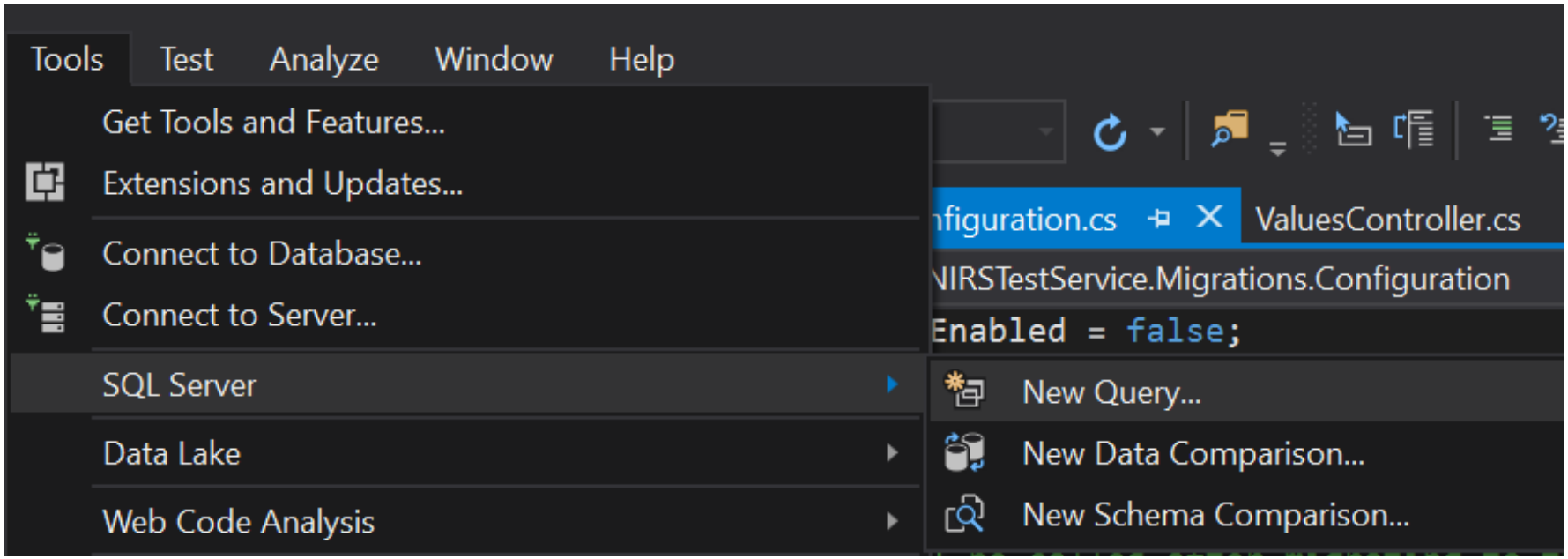
Bonus slides: Run your example Android App and create some data



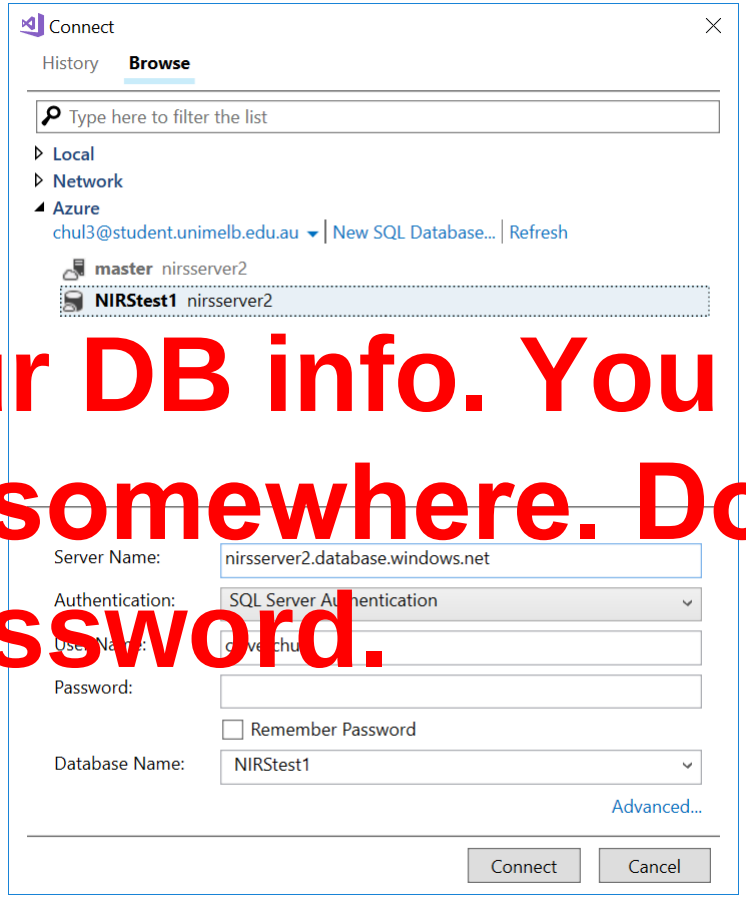
Check its source code to understand data management functions



Bonus slides: Check your SQL database using Visual Studio

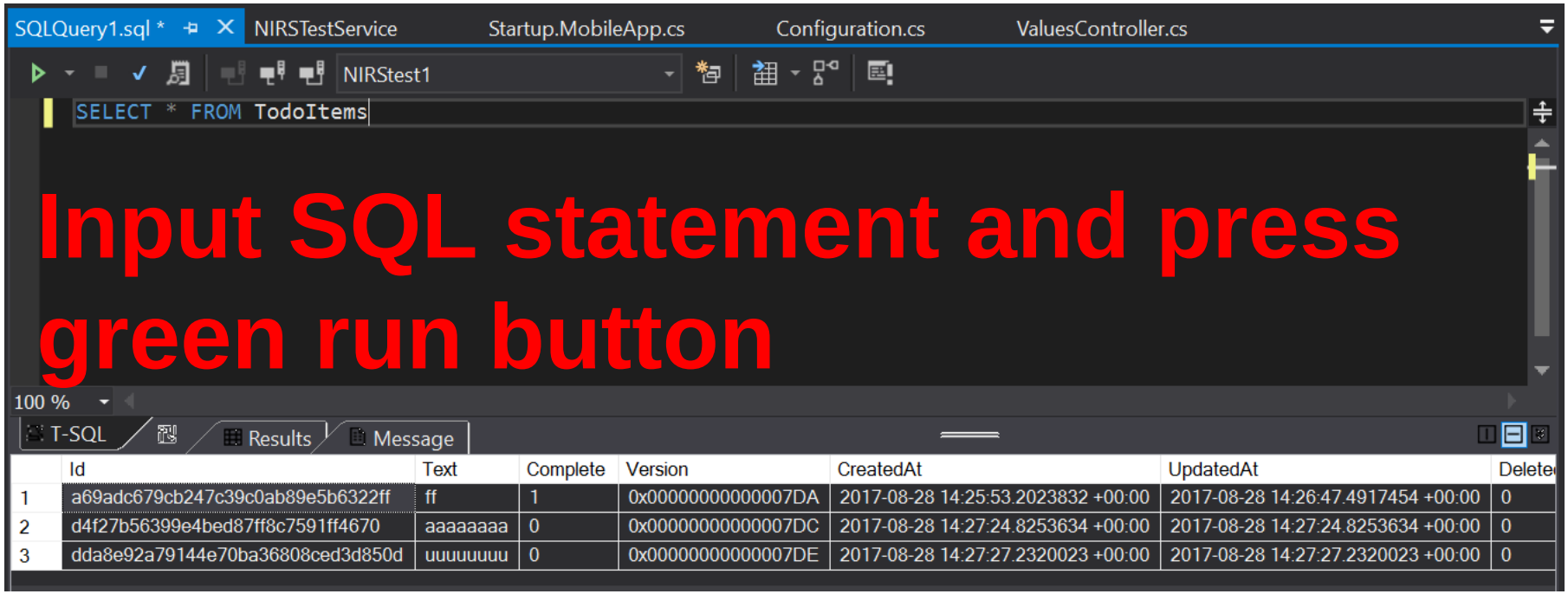


Bonus slides: Check your SQL database using Visual Studio



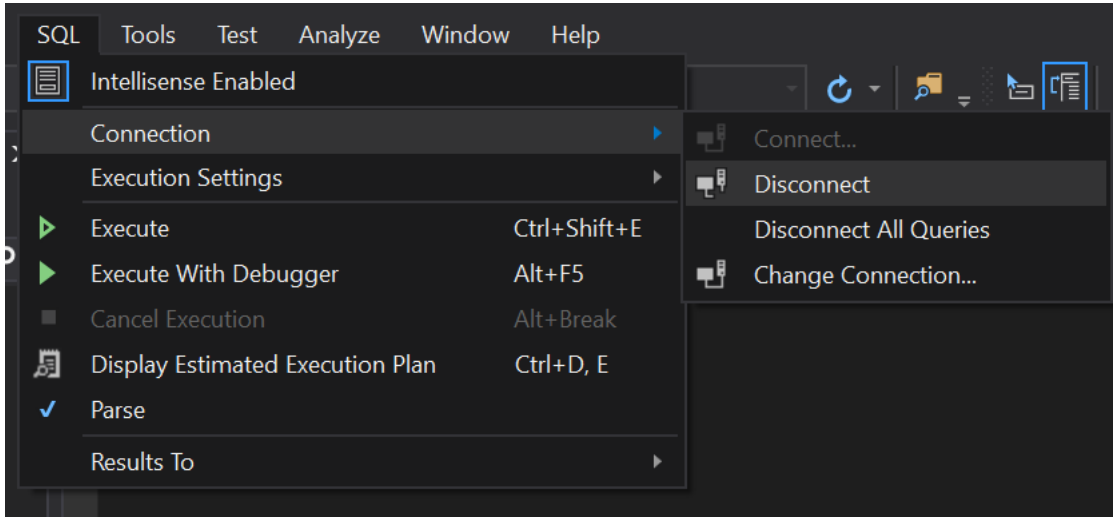
Input your DB info. You better record it somewhere. Don't forget password.

Bonus slides: Check your SQL database using Visual Studio



	Id	Text	Complete	Version	CreatedAt	UpdatedAt	Delete
1	a69adc679cb247c39c0ab89e5b6322ff	ff	1	0x00000000000007DA	2017-08-28 14:25:53.2023832 +00:00	2017-08-28 14:26:47.4917454 +00:00	0
2	d4f27b56399e4bed87ff8c7591ff4670	aaaaaaaa	0	0x00000000000007DC	2017-08-28 14:27:24.8253634 +00:00	2017-08-28 14:27:24.8253634 +00:00	0
3	dda8e92a79144e70ba36808ced3d850d	uuuuuuuu	0	0x00000000000007DE	2017-08-28 14:27:27.2320023 +00:00	2017-08-28 14:27:27.2320023 +00:00	0

Bonus slides: Check your SQL database using Visual Studio



Disconnect after use

Check All tables: SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE='BASE TABLE‘

Check all data in a table: SELECT * FROM TodoItems

Good luck and bug free

COMP 90018

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au