

# **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-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-dotnet-backend-how-to-use-server-sdk>**

# First, go to Azure portal



# App Services BackEnd

Microsoft Azure

Create a table API





Dashboard ▾ + New dashboard ✎ Edit dashboard 🔄 Share ↗ Fullscreen 🖼 Clone 🗑 Delete

+ New









- Dashboard
- All resources
- Resource groups
- App Services
- Function Apps
- SQL databases
- Azure Cosmos DB
- Virtual machines
- Load balancers

All resources  
ALL SUBSCRIPTIONS

Refresh

	NIRSTest	App Service
	NIRStest1	SQL database
	nirsserver2	SQL server
	Test	App Service plan

Quickstart tutorials

-  [Windows Virtual Machines](#)   
Provision Windows Server, SQL Server, SharePoint VMs
-  [Linux Virtual Machines](#)   
Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs
-  [App Service](#)   
Create Web Apps using .NET, Java, Node.js, Python, PHP
-  [Functions](#)   
Process events with a serverless code architecture

**Get a DB and App service**



# App Services BackEnd

The screenshot shows the Microsoft Azure App Services console. The top navigation bar includes the Microsoft Azure logo, the path 'App Services > NIRSTest - Quickstart', and a user profile 'chul3@student.unime... THE UNIVERSITY OF MELBOU...'. The left sidebar shows the 'App Services' section with a search bar and a list of subscriptions. The main content area is divided into three panes. The left pane shows the 'NIRSTest' app. The middle pane shows the 'DEPLOYMENT' section with 'Quickstart' selected. The right pane shows the 'GENERAL' settings, where the 'Android' platform is highlighted with a red box. A large red text overlay 'Select Android' is positioned over the 'Android' option.

Microsoft Azure App Services > NIRSTest - Quickstart

App Services  
The University of Melbourne

+ Add Columns Refresh

Subscriptions: Microsoft Azure Sponsorship 2

Filter by name...

1 items

NAME

NIRSTest

Search (Ctrl+ /)

Diagnose and solve problems

DEPLOYMENT

Quickstart

Deployment credentials

Deployment slots

Deployment options

Continuous Delivery (Preview)

SETTINGS

Application settings

Filter settings

GENERAL

iOS (Objective-C)

iOS (Swift)

**Android**

Windows (C#)

Windows 8.1 (C#)

Xamarin.Android

Xamarin.iOS

Xamarin.Forms

Cordova

Select Android






# App Services BackEnd

Android

Quick start

1


Connect a database



You already have a data connection

2

Create a table API



To store data in your backend, you need a table. Pick a backend language below and create a TodoItem table API.

Backend language:

C#

Download

C# or Node.js

(This slides use C#)




# App Services BackEnd

Android

Quick start

1


Connect a database



You already have a data connection

2

Create a table API



To store data in your backend, you need a table. Pick a backend language below and create a TodoItem table API.

Backend language:

C#

Download

**Download and  
Open in Visual Studio**



# App Services BackEnd

Android

Quick start

3

Configure your client application

CREATE A NEW APP

CONNECT AN EXISTING APP

Connect your app

If you are using Eclipse, see instructions on [Migrating to Android Studio](#)

Alternatively, you can [download the Mobile Services SDK for Android](#) (Android 2.2+ supported), extract the SDK files, and copy the .jar files into the lib folder of your app.

For Android Studio, add the following lines to the project's Gradle.build file:

```
buildscript {  
    repositories {  
        jcenter()  
    }  
}
```

And add the following to the app's Gradle build file with your desired SDK version plugged in (you can find the latest versions [here](#)):

**This is Android  
Connects to Azure**



# App Services BackEnd

3

Configure your client application

CREATE A NEW APP

CONNECT AN EXISTING APP

On a Mac or Windows PC: [Download Android Studio](#)

Download your personalized Android project, extract it into a folder, and then in Android Studio, select Import project (Eclipse ADT, Gradle, etc.). The app is pre-configured to work with your hosted mobile backend.

Download

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

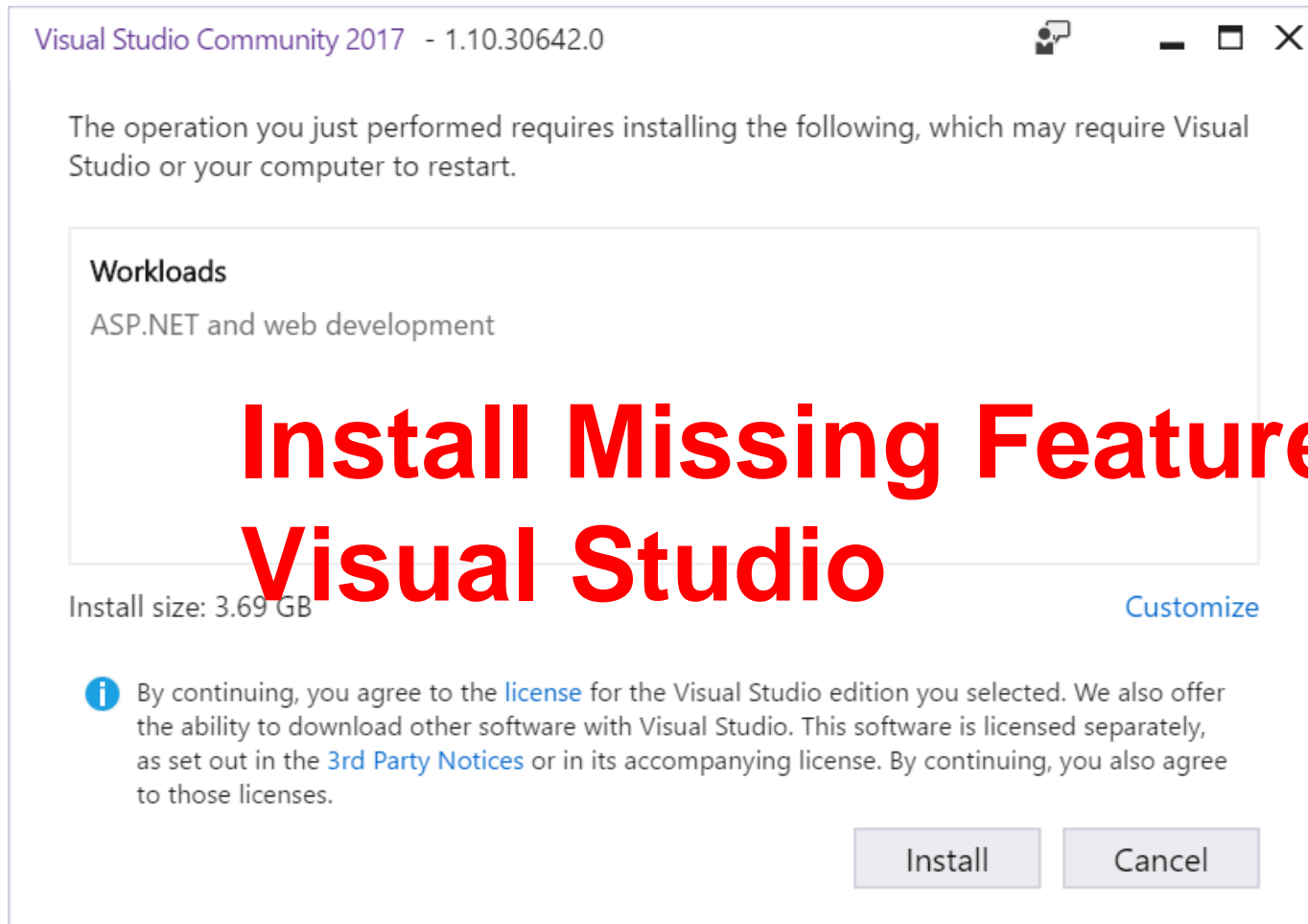
Run the Android project to start working with data in your mobile backend.



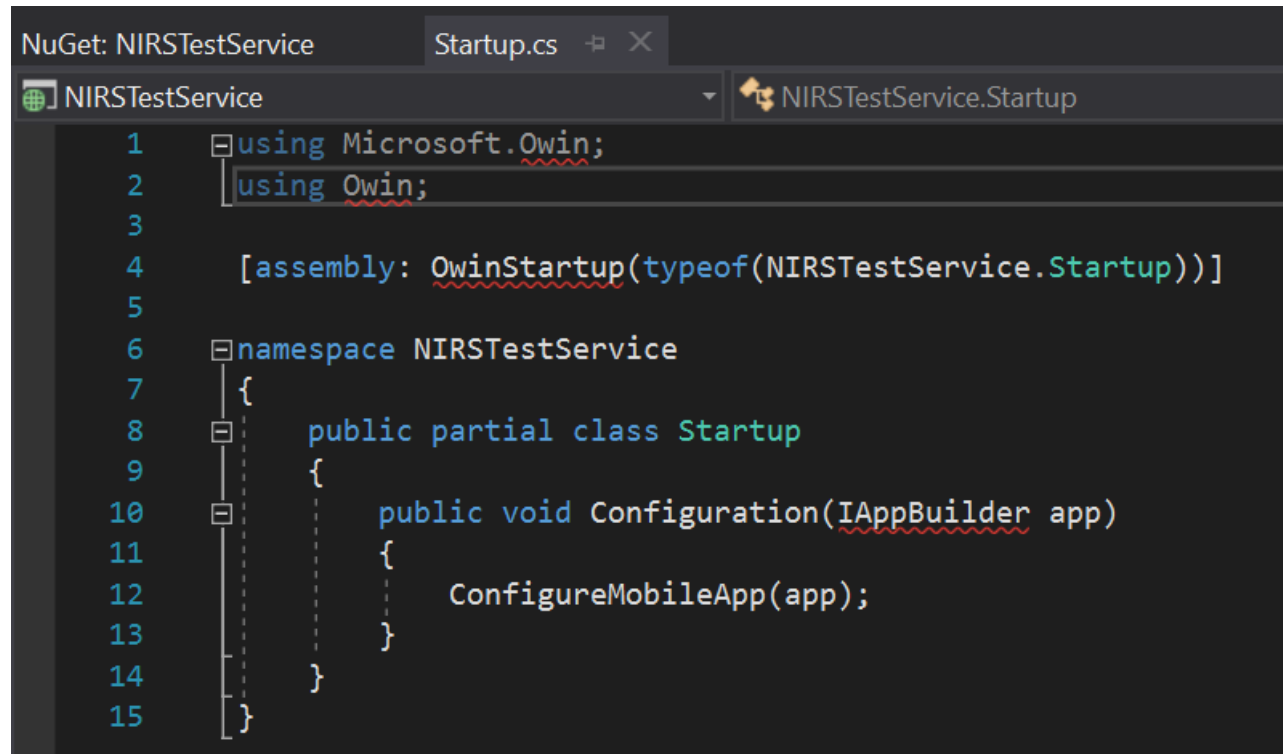
**Then, open the download solution file (Not Android example app) using Visual Studio  
(Log in with your account, same as Azure)**

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

# App Services BackEnd



# Install SDK

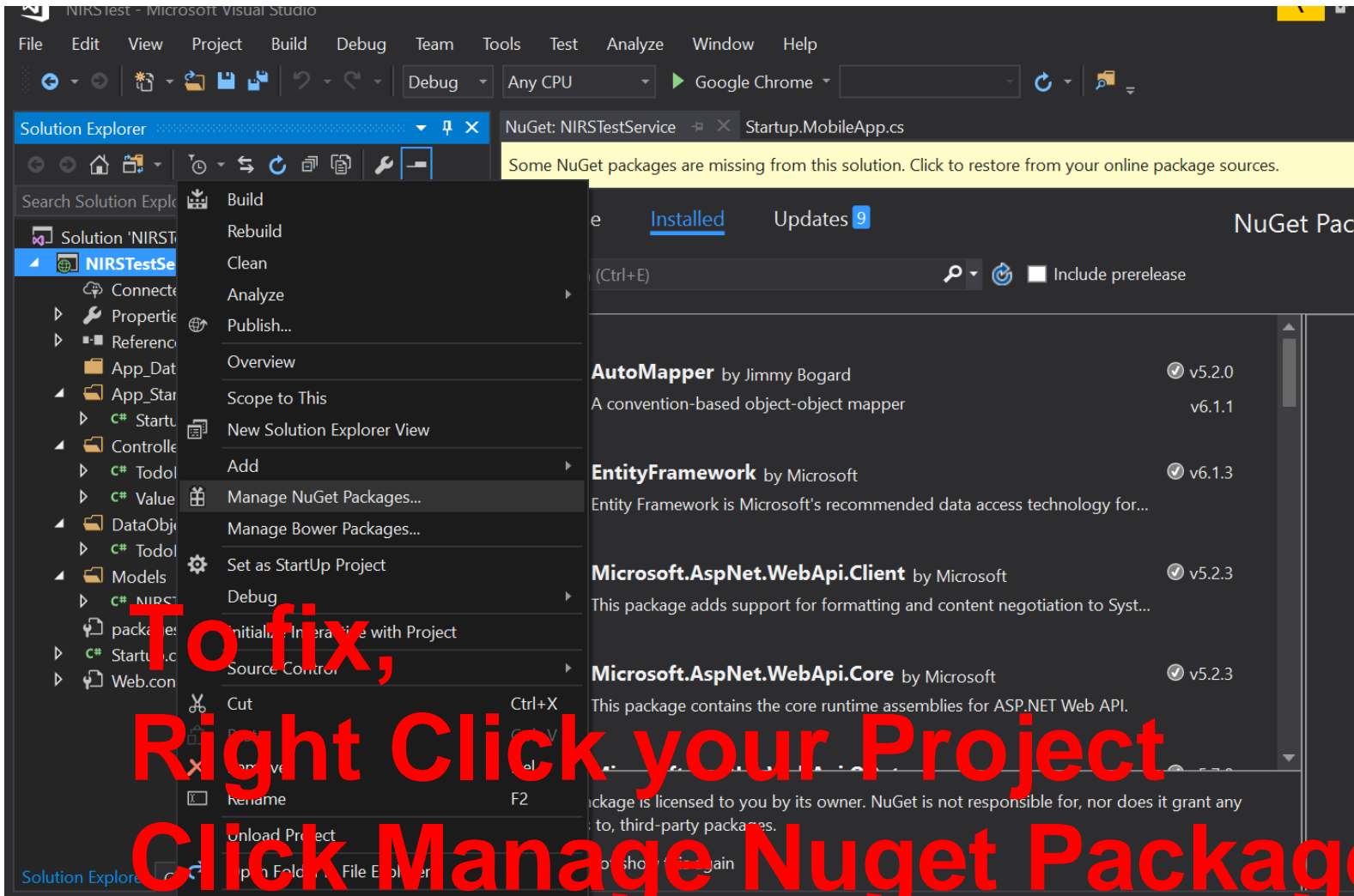


```
NuGet: NIRTestService Startup.cs
NIRTestService
1  using Microsoft.Owin;
2  using Owin;
3
4  [assembly: OwinStartup(typeof(NIRTestService.Startup))]
5
6  namespace NIRTestService
7  {
8      public partial class Startup
9      {
10         public void Configuration(IAppBuilder app)
11         {
12             ConfigureMobileApp(app);
13         }
14     }
15 }
```

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



# Install SDK



Some NuGet packages are missing from this solution. Click to restore from your online package sources.

**To fix,**  
**Right Click your Project**  
**Click Manage Nuget Packages**

**NuGet Packages**

Package Name	Version
AutoMapper	v5.2.0
EntityFramework	v6.1.1
Microsoft.AspNet.WebApi.Client	v5.2.3
Microsoft.AspNet.WebApi.Core	v5.2.3



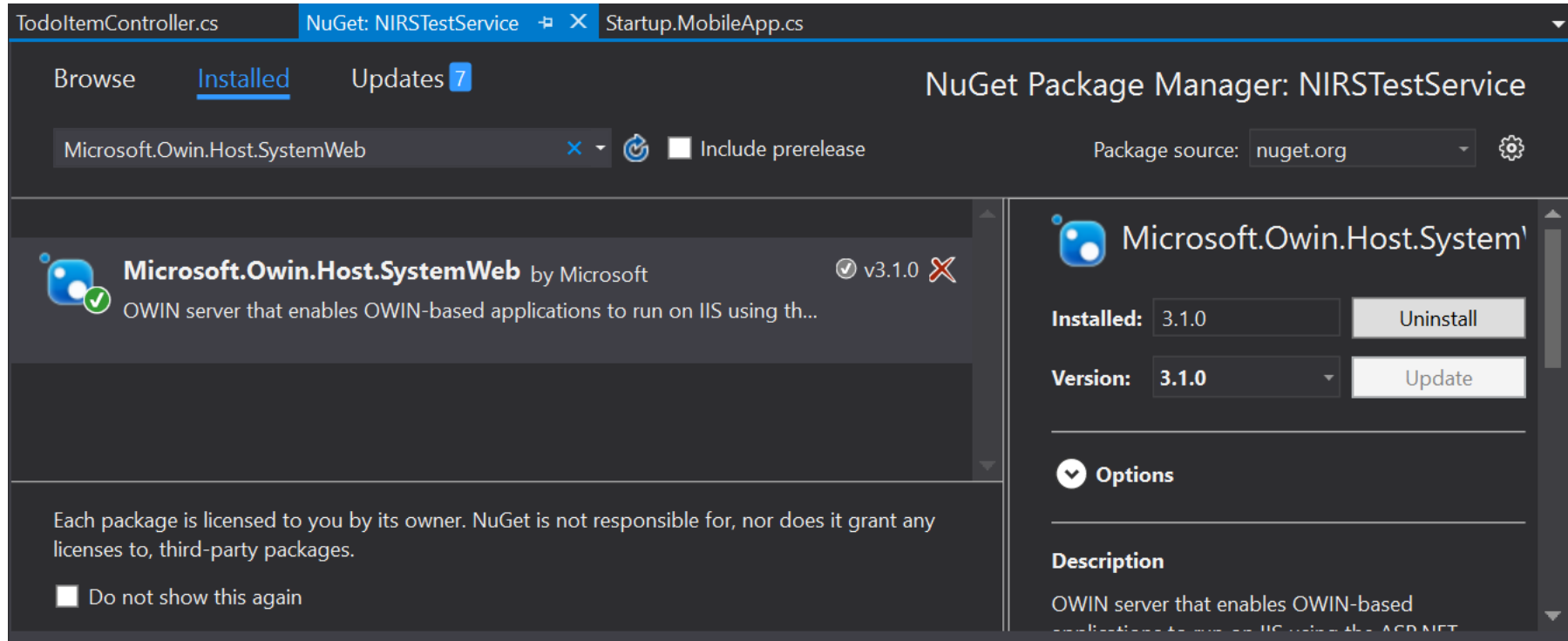
# Install SDK

**Install whatever you need.**

**Depends on your Visual Studio  
State**



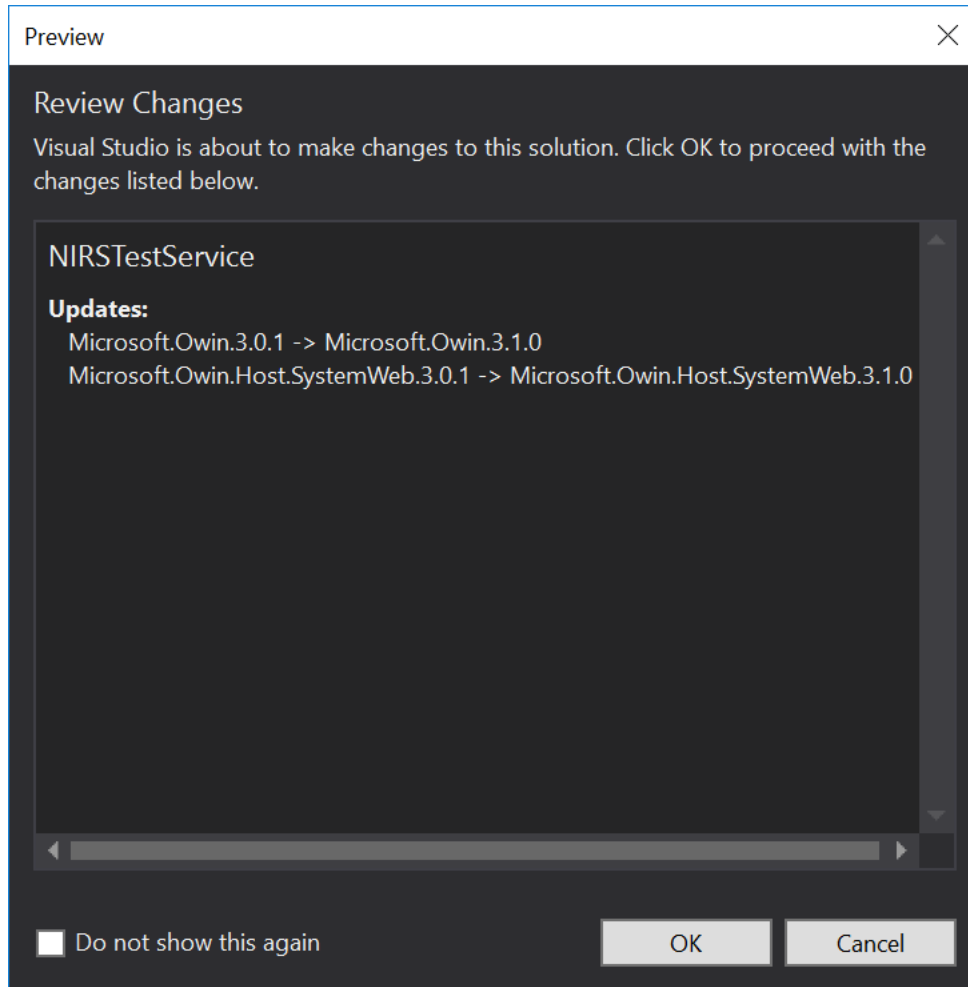
# Install SDK



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

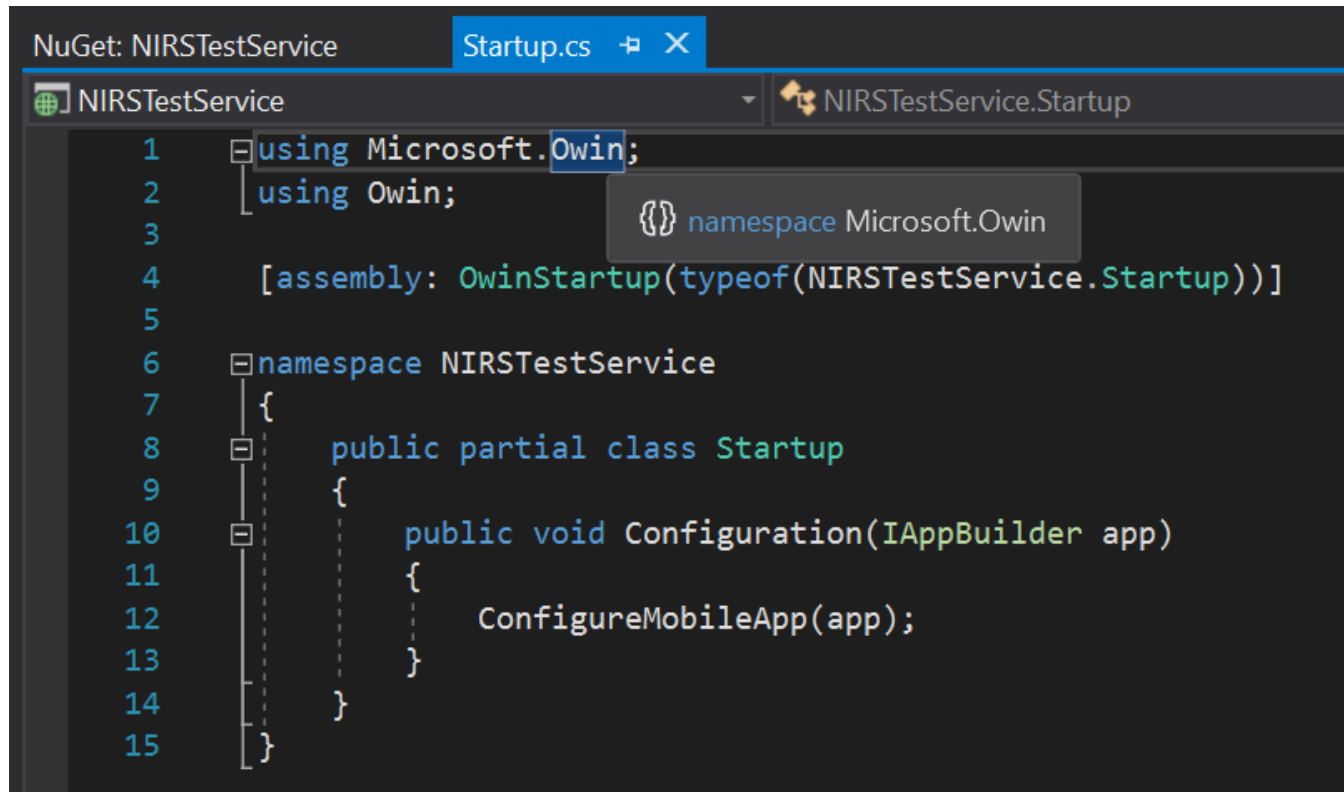


# Install SDK



**Apply  
changes**

# Install SDK



```
NuGet: NIRTestService Startup.cs X
NIRTestService NIRTestService.Startup
1 using Microsoft.Owin;
2 using Owin;
3
4 [assembly: OwinStartup(typeof(NIRTestService.Startup))]
5
6 namespace NIRTestService
7 {
8     public partial class Startup
9     {
10         public void Configuration(IAppBuilder app)
11         {
12             ConfigureMobileApp(app);
13         }
14     }
15 }
```

**Then everything should work**

# Initialise BackEnd

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



# Initialise BackEnd

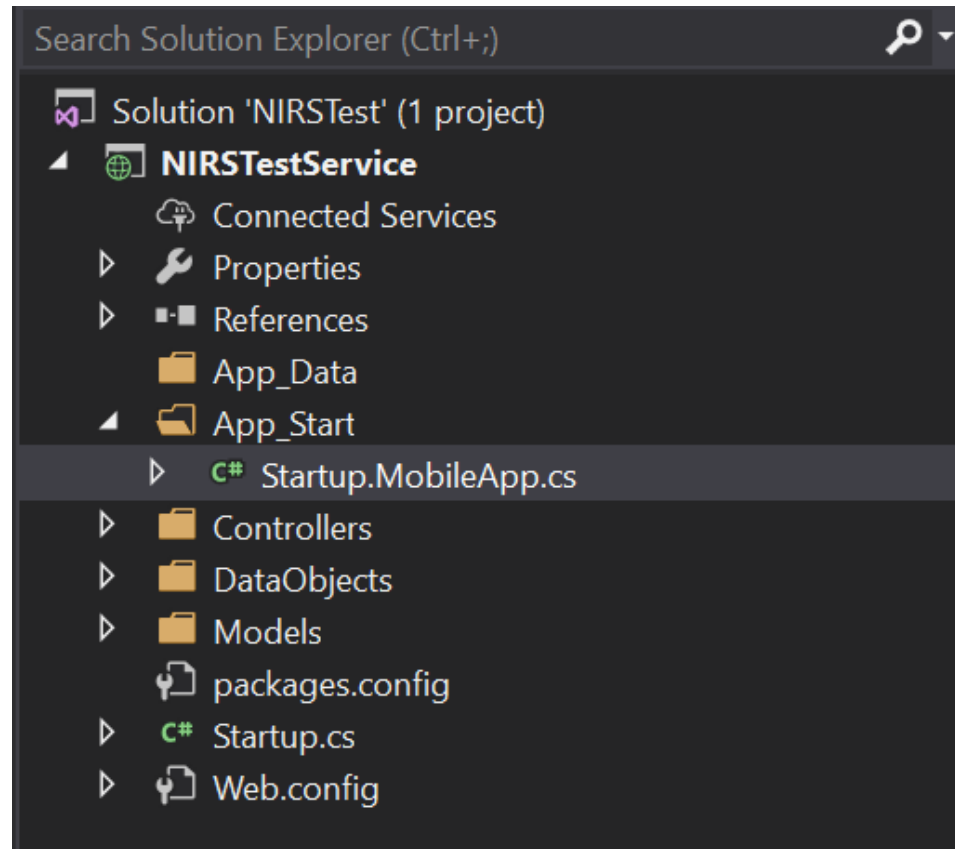
## Initialize the server project

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



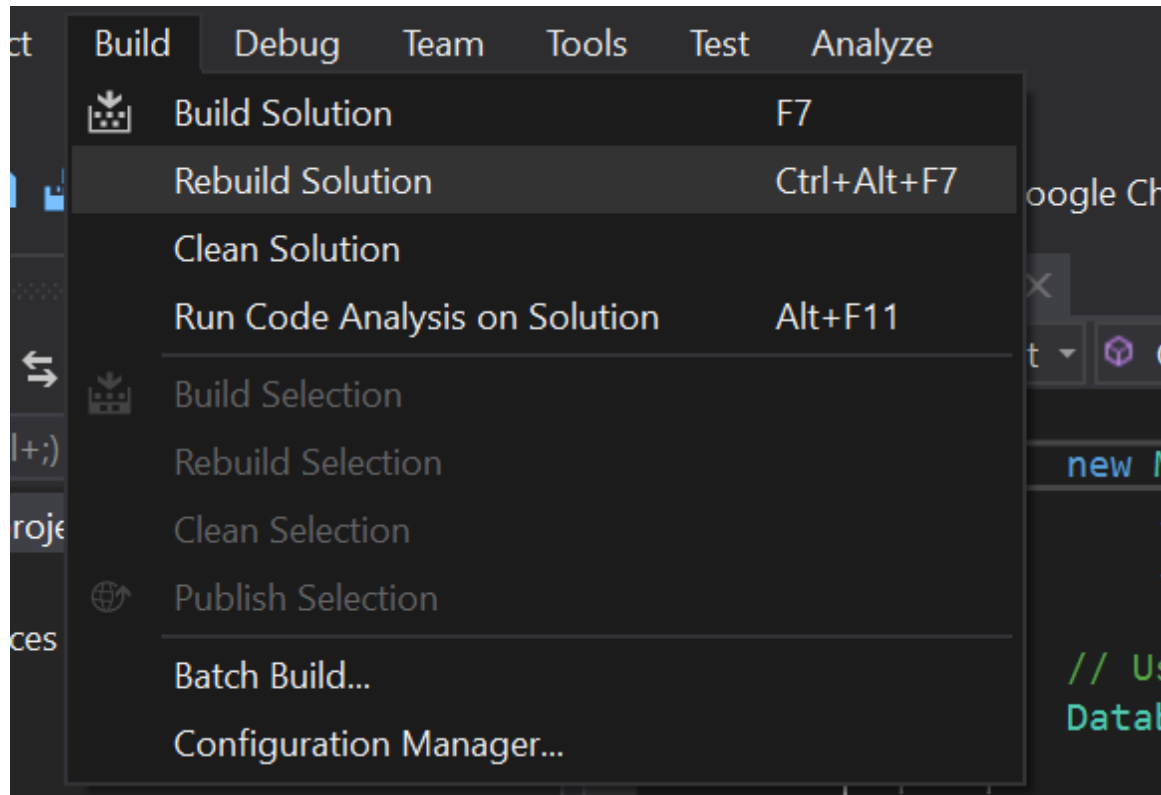
**You already have it here.**



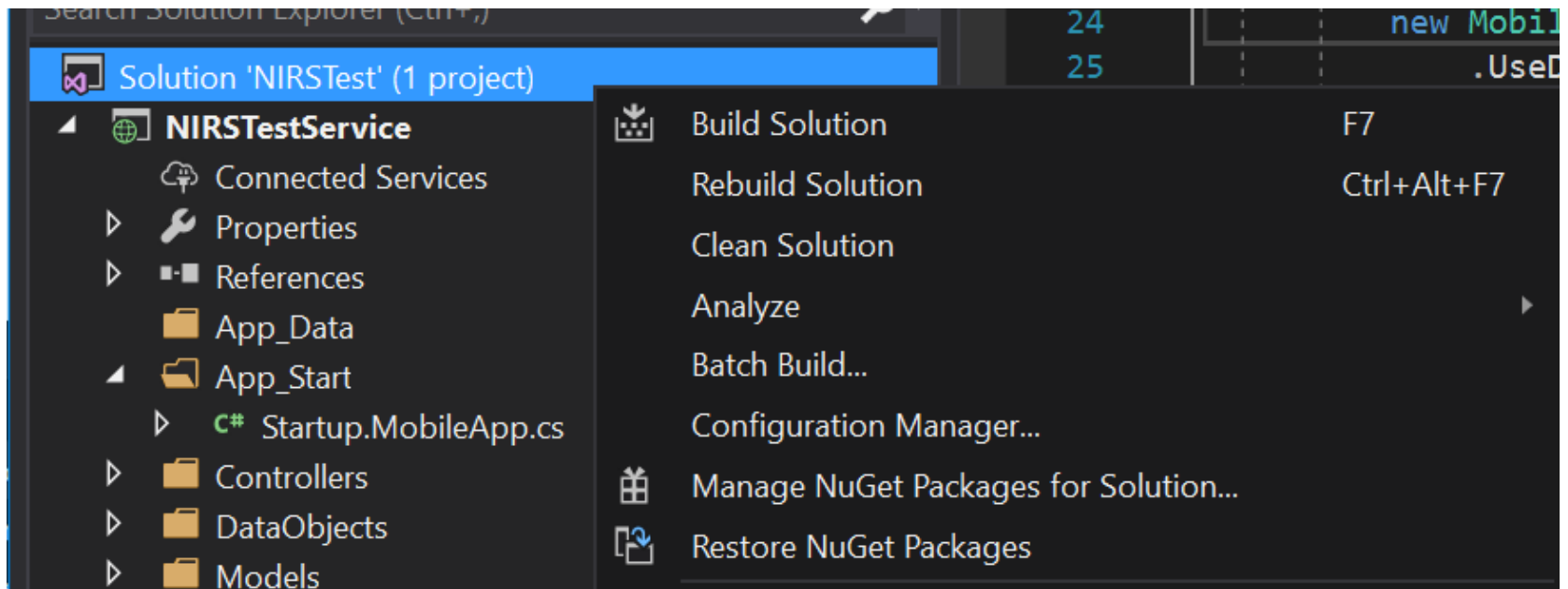


# How to publish your BackEnd online?

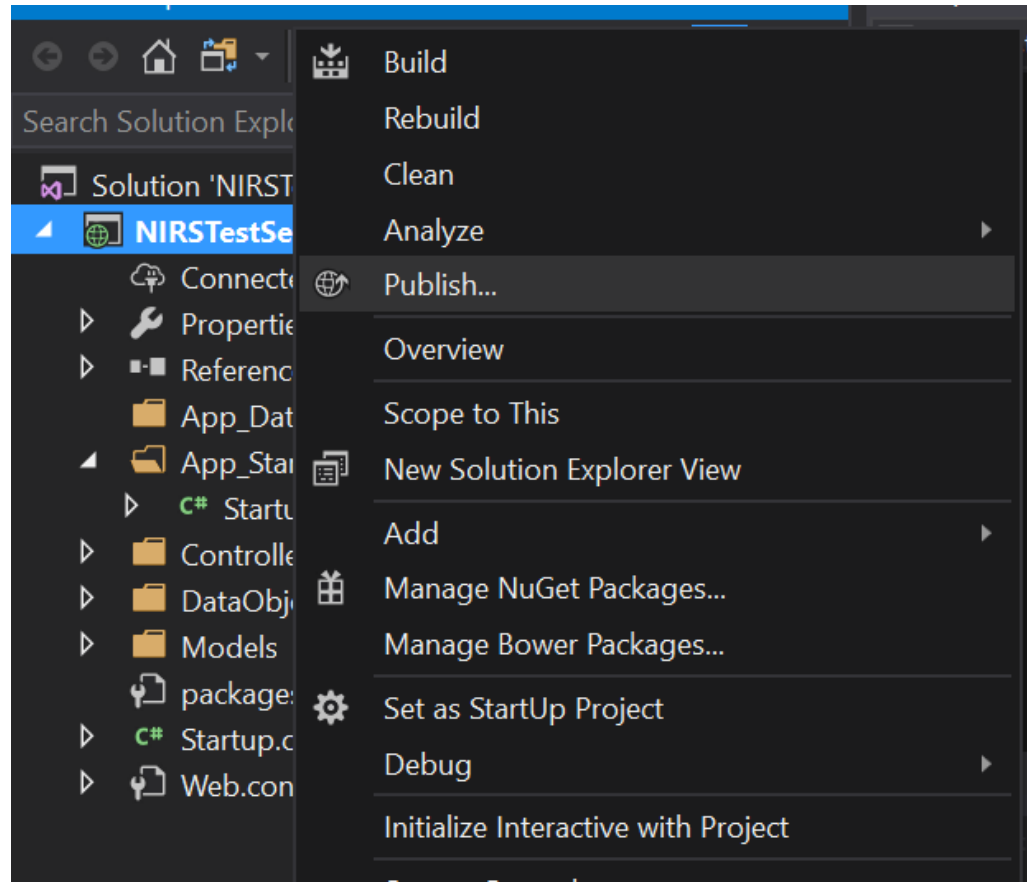
# 1. Rebuild



## 2. Restore NuGet



# 3. Publish



# 4. Select Existing App Service


The screenshot shows the Visual Studio Publish dialog box for a project named 'NIRSTestService'. The 'Startup.MobileApp.cs' file is selected. The 'Publish' tab is active in the left sidebar. The main area is titled 'Publish' and contains the text 'Publish your app to Azure or another host. [Learn more](#)'. There are three options for publishing: 'Microsoft Azure App Service' (selected), 'IIS, FTP, etc', and 'Folder'. Below these options are two radio buttons: 'Create New' (selected) and 'Select Existing'. A 'Publish' button is located at the bottom right.


NIRSTestService Startup.MobileApp.cs NuGet: NIRSTestService Startup.cs


Overview  
Connected Services  
**Publish**

**Publish**

Publish your app to Azure or another host. [Learn more](#)

  
Microsoft Azure  
App Service

  
IIS, FTP, etc

  
Folder

☒ Create New  
☐ Select Existing

**Publish**

# 4. Select Existing App Service


The screenshot shows the Visual Studio Publish dialog for a project named 'NIRSTestService'. The 'Publish' tab is selected in the left sidebar. The main area displays the 'Publish' title and a description: 'Publish your app to Azure or another host. [Learn more](#)'. Below this, there are three selectable options, each with an icon and a label: 'Microsoft Azure App Service' (cloud icon), 'IIS, FTP, etc' (document icon), and 'Folder' (folder icon). At the bottom left, there are two radio buttons: 'Create New' (selected) and 'Select Existing'. At the bottom right, there is a 'Publish' button. The top of the dialog shows the project name 'NIRSTestService' and the file 'Startup.MobileApp.cs'.


NIRSTestService Startup.MobileApp.cs NuGet: NIRSTestService Startup.cs


Overview  
Connected Services  
**Publish**

**Publish**

Publish your app to Azure or another host. [Learn more](#)

  
Microsoft Azure App Service

  
IIS, FTP, etc


  
Folder

☒ Create New  
☐ Select Existing

**Publish**

# 5. Find your Azure App Service

**App Service**  
Host your web and mobile applications, REST APIs, and more in Azure

 The University of Melbourne  
chul3@student.unimelb.edu.au

Subscription

Microsoft Azure Sponsorship 2

View

Resource Group

Search

▲

📁

NIRSTest

▲

+

NIRSTest

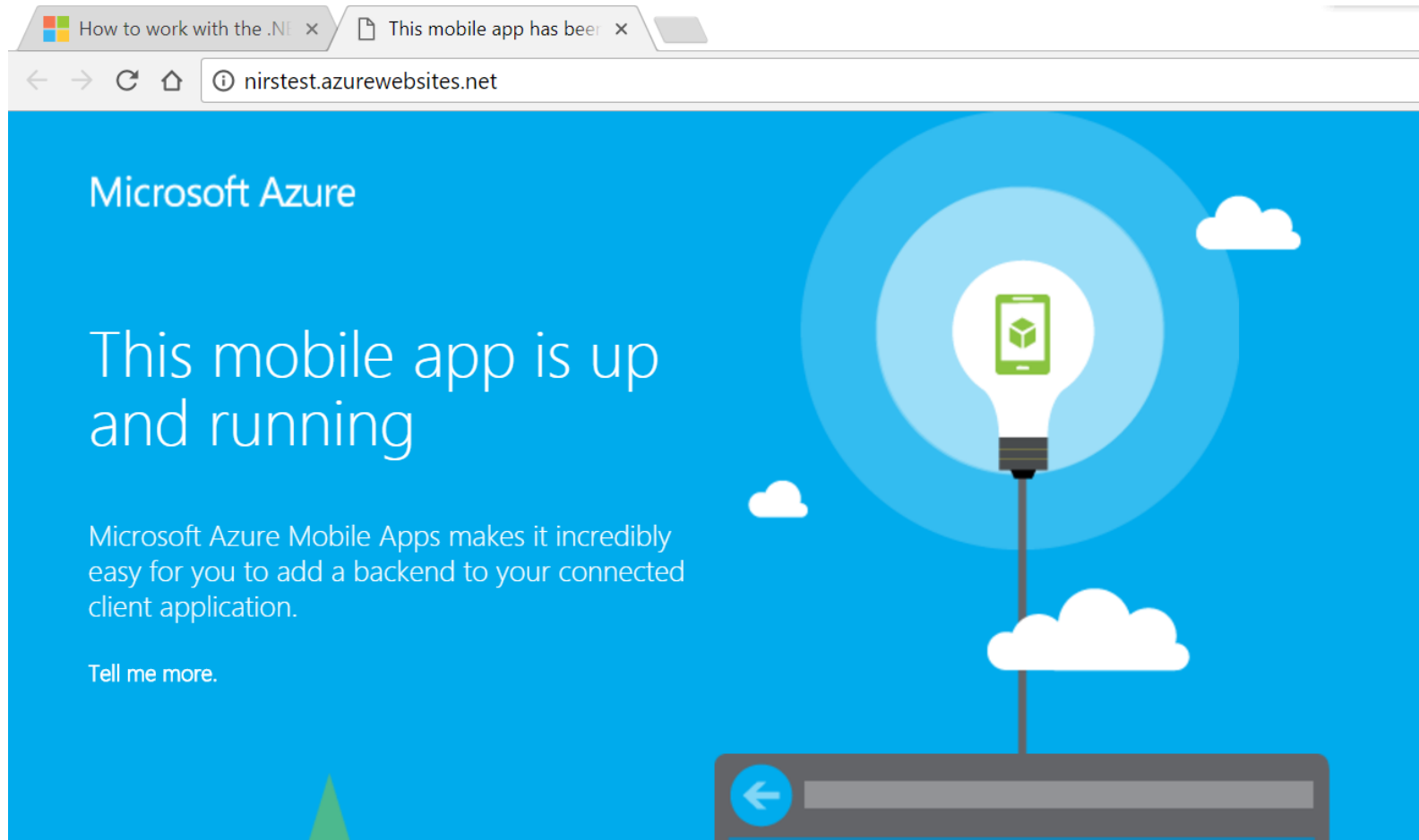
📁

Deployment Slots

OK

Cancel

# 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)**

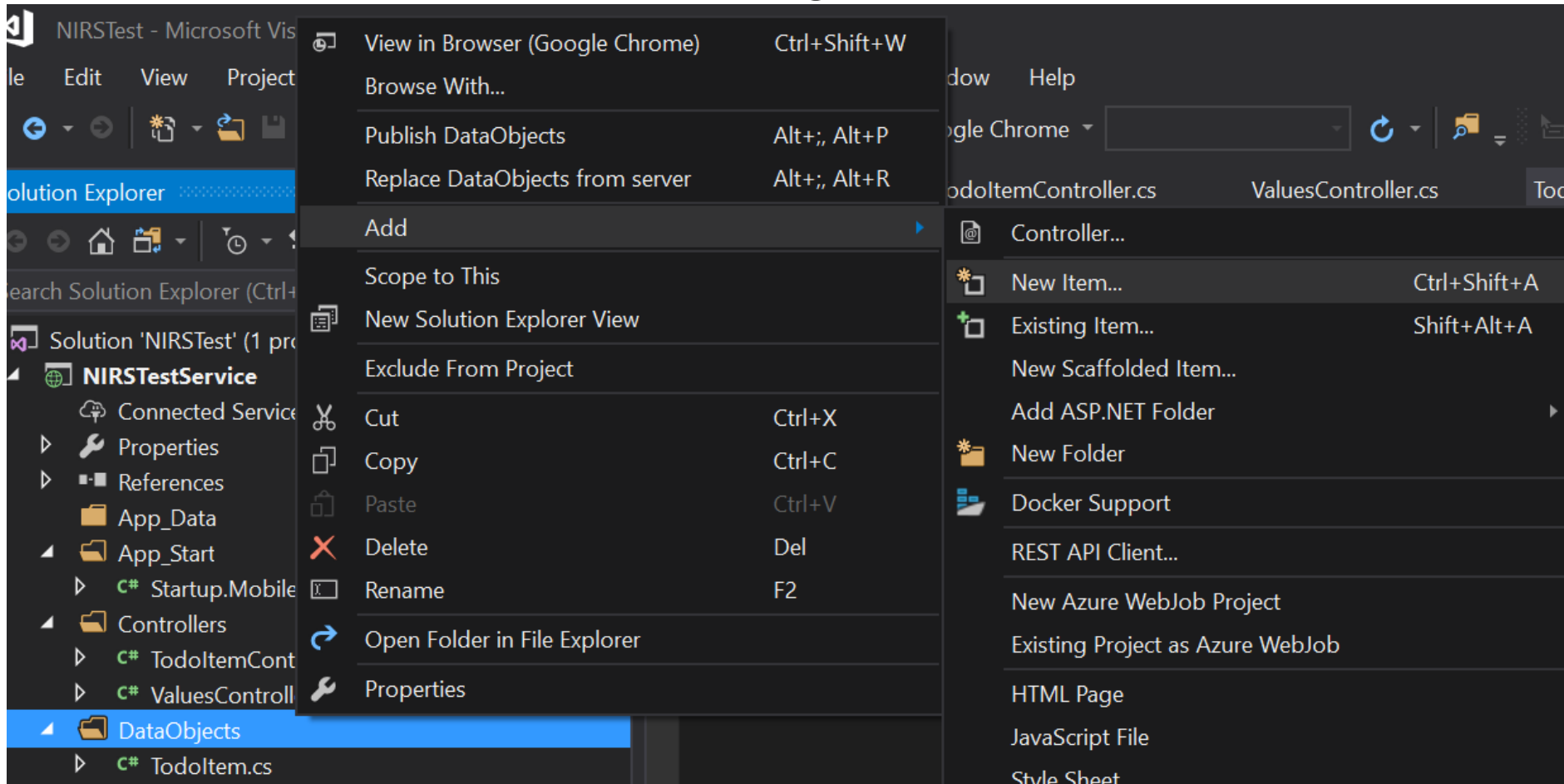
**a Table Controller**

**a Code-First Migration**

# Data Transfer Object

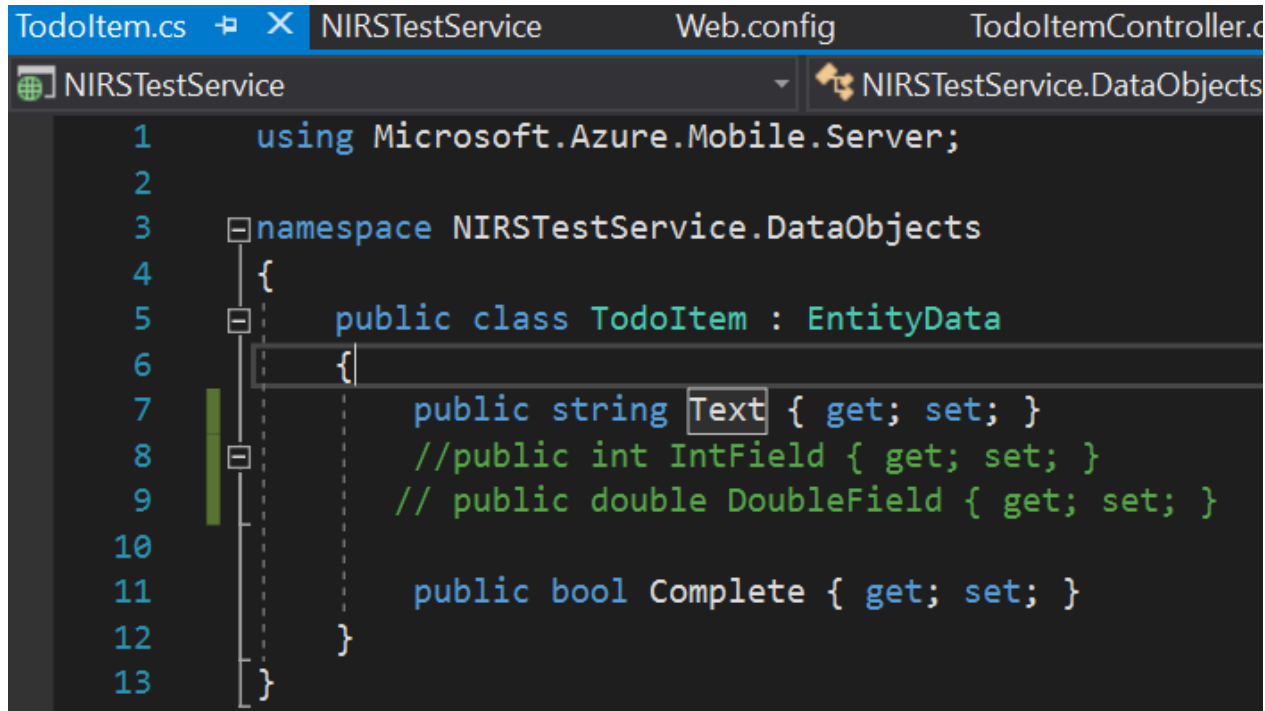
**This is the object design  
same as your database  
schema**

# Data Transfer Object



**If you want to add, use C# class**

# Data Transfer Object



```
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 }
```

**Todoltem 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**

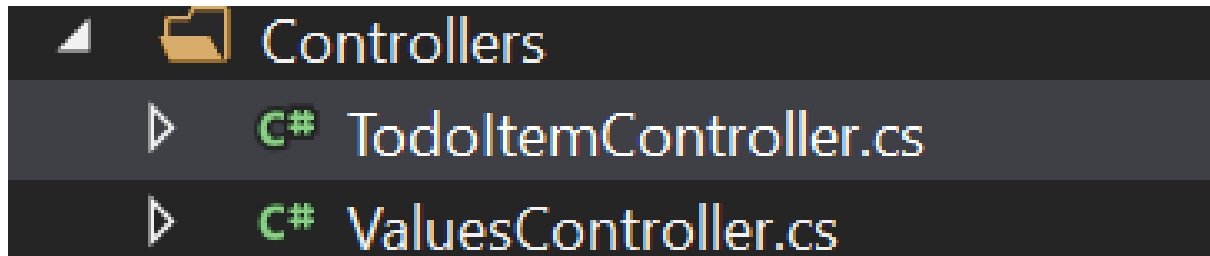
# To use a Table Controller, we need to create:

- ✓ a Data Transfer Object  
(DTO)

**a Table Controller**

**a Code-First Migration**

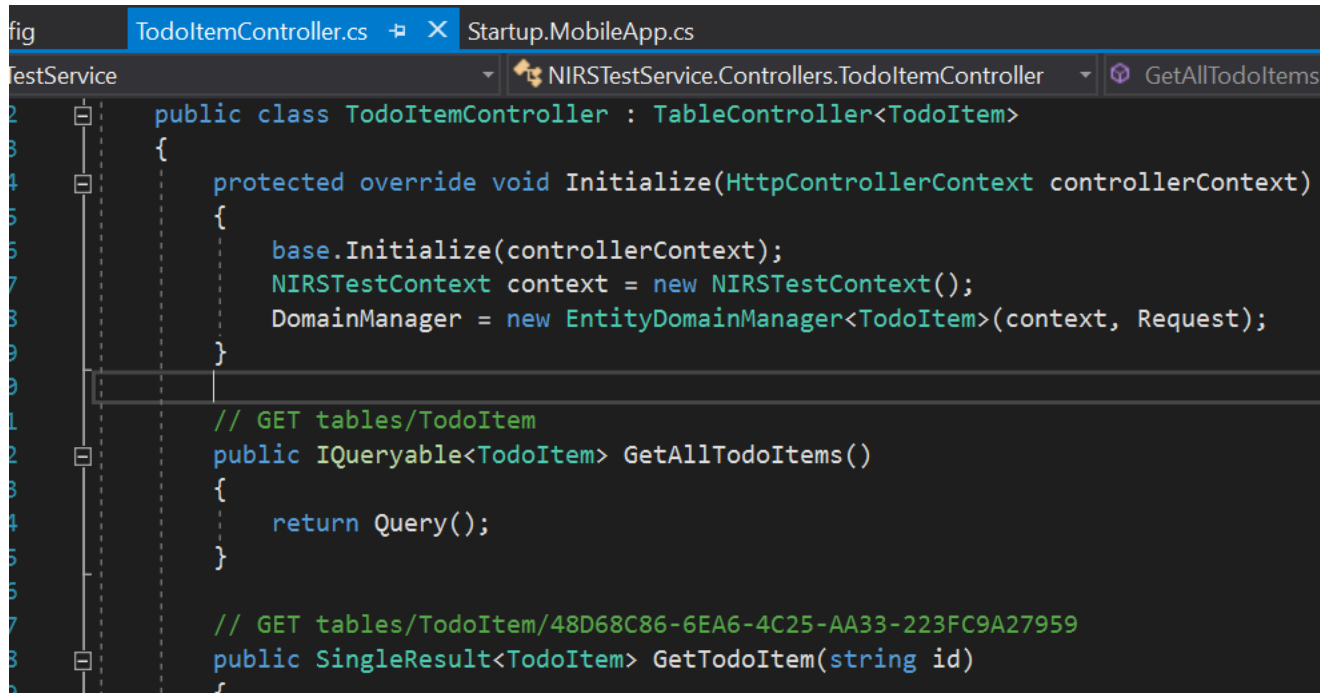
# Table Controller



**TodoltemController is an existing example**



# Table Controller



```
fig  TodoItemController.cs  Startup.MobileApp.cs
testService  NIRSTestService.Controllers.TodoItemController  GetAllTodoItems

public class TodoItemController : TableController<TodoItem>
{
    protected override void Initialize(HttpControllerContext controllerContext)
    {
        base.Initialize(controllerContext);
        NIRSTestContext context = new NIRSTestContext();
        DomainManager = new EntityDomainManager<TodoItem>(context, Request);
    }

    // GET tables/TodoItem
    public IQueryable<TodoItem> GetAllTodoItems()
    {
        return Query();
    }

    // GET tables/TodoItem/48D68C86-6EA6-4C25-AA33-223FC9A27959
    public SingleResult<TodoItem> GetTodoItem(string id)
    {
    }
}
```

**TodoltemController 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**

Publish

?

×

 Publish

Connection

Settings

**NIRSTest - Web Deploy \***

Configuration: Release

File Publish Options


Databases

NIRSTestContext (MS\_TableConnectionString)

Data Source=tcp:nirserver2.database.windows.net,1433;Initial Catalog=NIRS

☒ Use this connection string at runtime (update destination web.config)

☐ Execute Code First Migrations (runs on application start)

 In order to publish a Code First model, Code First Migrations should be used.  
[Learn more about this](#)

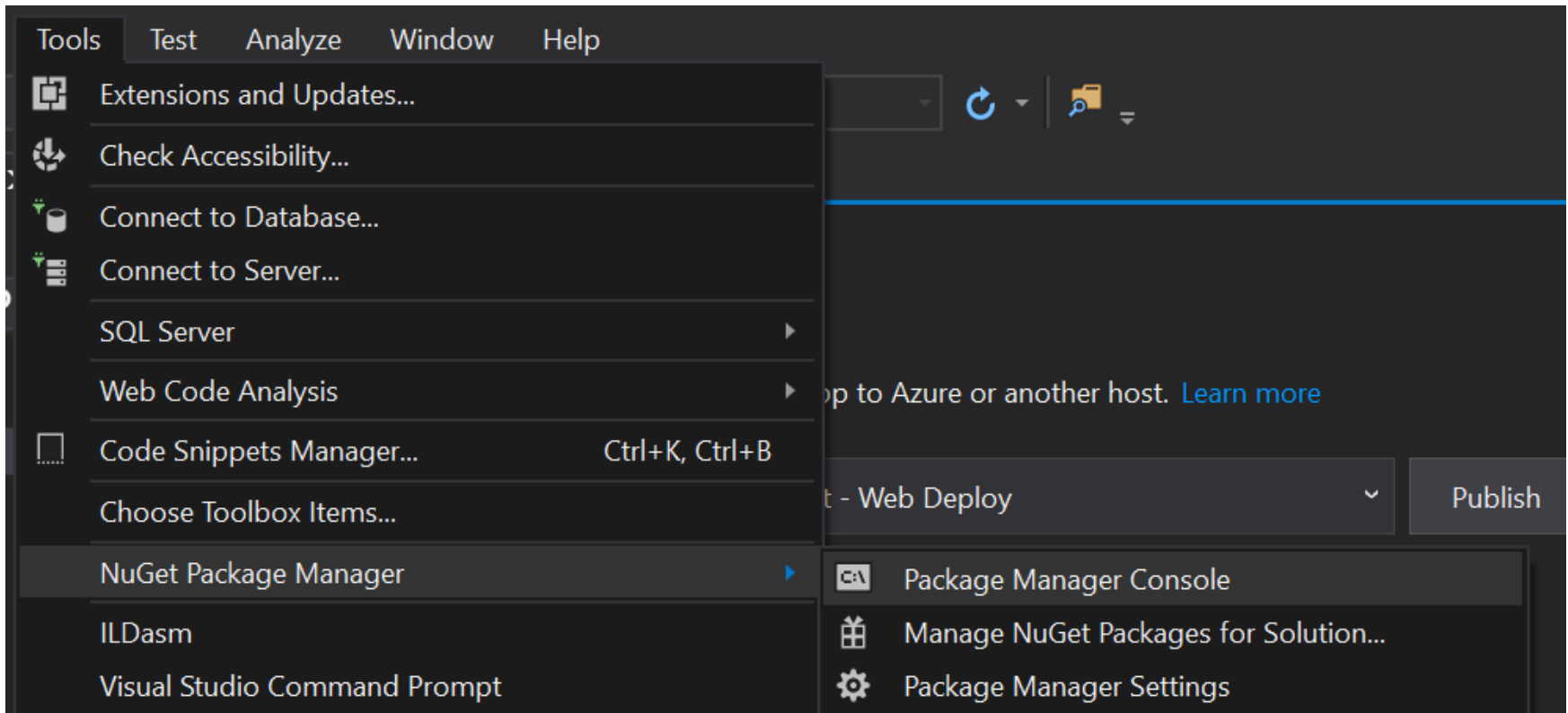
< Prev

Next >

Save

Cancel

**By default, it is disabled**



# Open this Console

## Package Manager Console

Package source:   Default project:

Each package is licensed to you by its owner. NuGet is not responsible for packages. Some packages may include dependencies which are governed by add to determine any dependencies.


Package Manager Console Host Version 4.2.0.2457

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

PM>

# You will see this

## Package Manager Console

Package source: All  Default project: NIRSTestService

Each package is licensed to you by its owner. NuGet is not responsible for packages. Some packages may include dependencies which are governed by add to determine any dependencies.

Package Manager Console Host Version 4.2.0.2457

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

PM>

# 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|
```

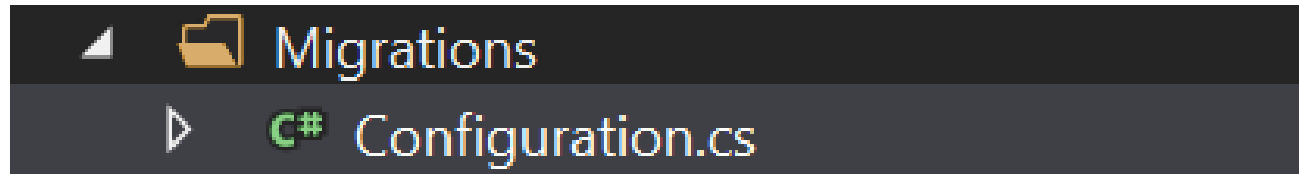
```
PM>
```

**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**

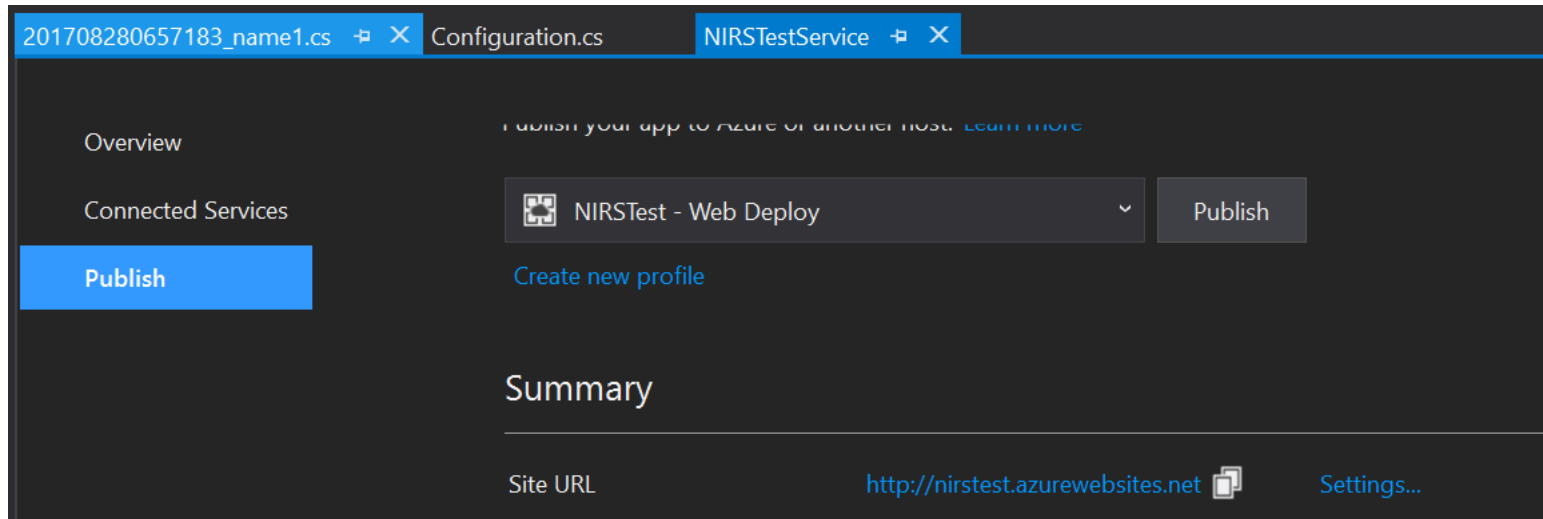
Check:

<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 <40> <40> <40> ]> 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 <40> <40> <40> ]> |
```

**Well, Chu does not have a local one**



# Now go to Publish Settings



Publish

Connection

Settings

### NIRSTest - Web Deploy \*

Configuration: Release

#### File Publish Options

#### Databases

#### NIRSTestContext (MS\_TableConnectionString)

Data Source=tcp:nirserver2.database.windows.net,1433;Initial Catalog=NIRS

☒ Use this connection string at runtime (update destination web.config)

☒ Execute Code First Migrations (runs on application start)

< Prev

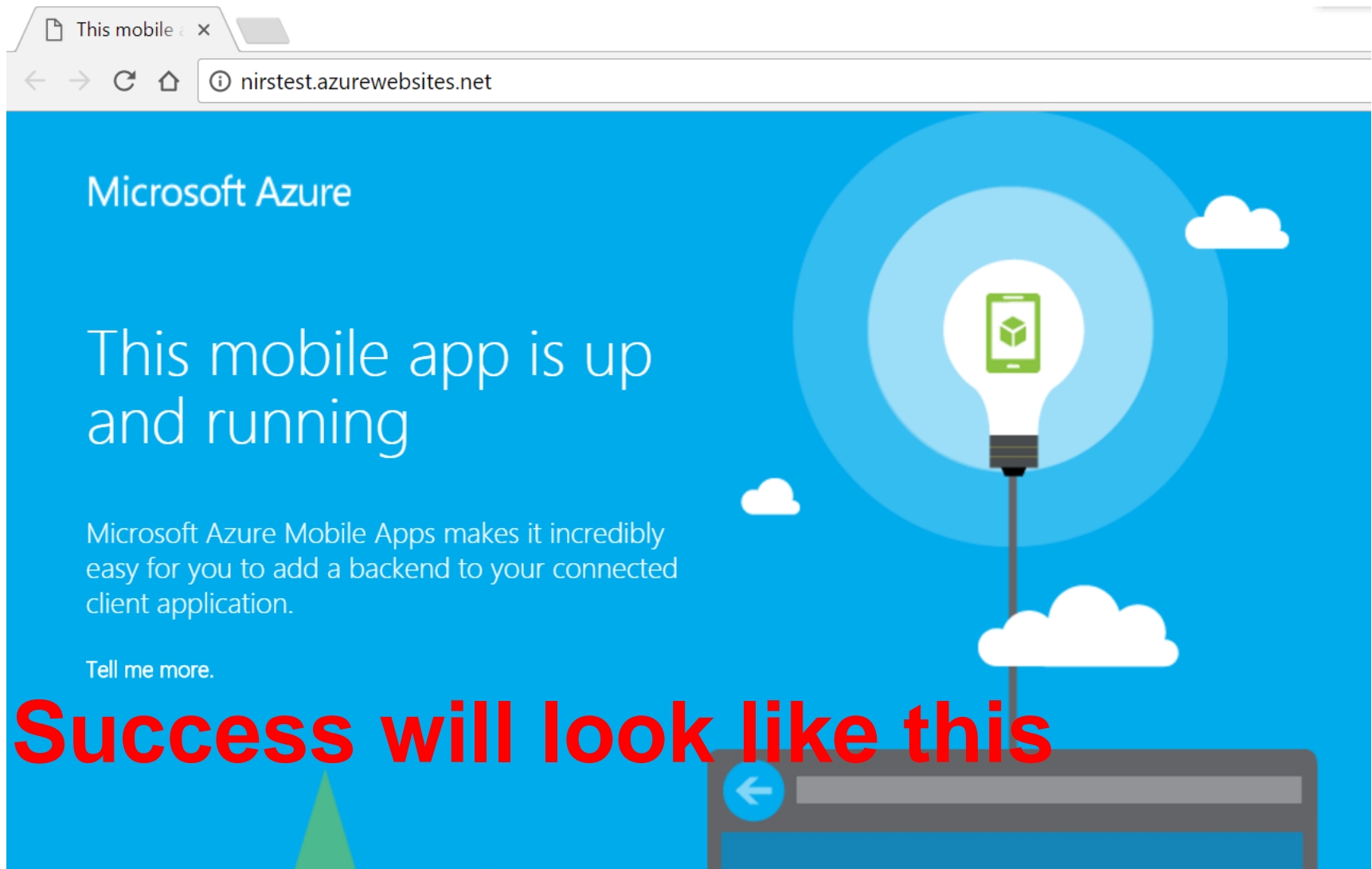
Next >

Save

Cancel

**Tick Code First Mig and Save  
No press Cancel...**





# 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**

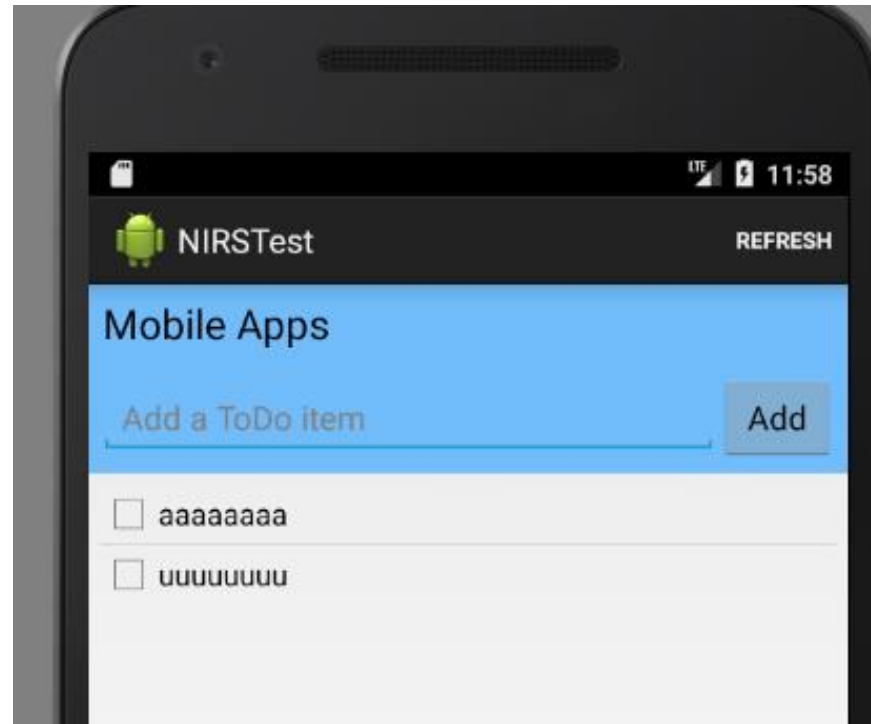
**COMP 90018**

**Tutorial on Android Development**

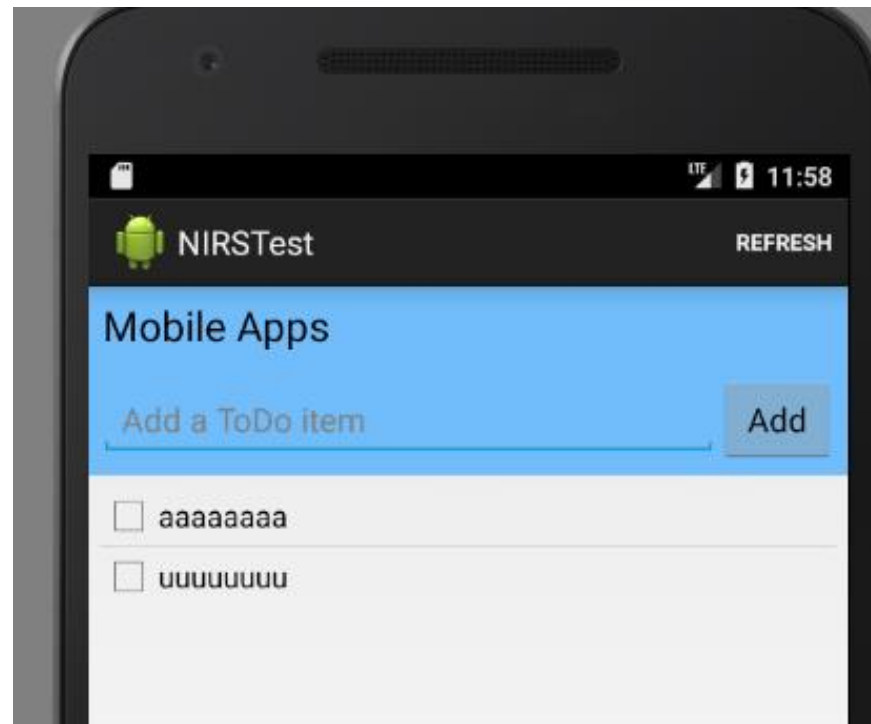
**Chu Luo, Eman Bin Khunayn**

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

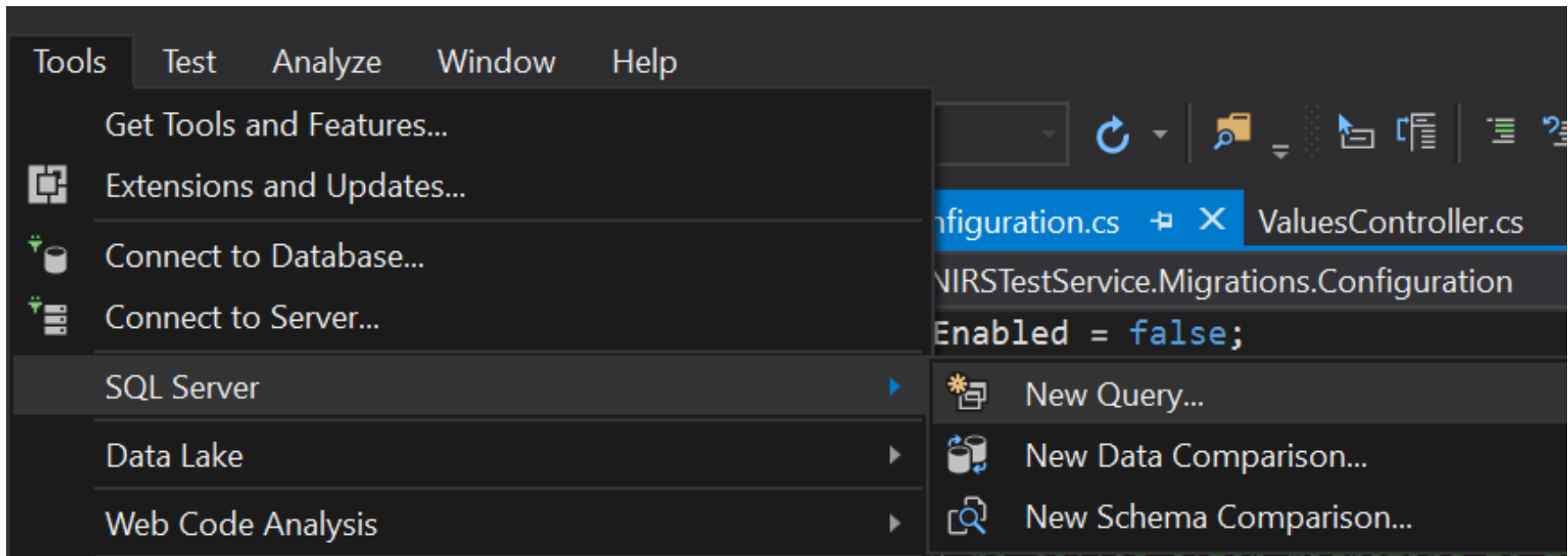
# Bonus slides: Run your example Android App and create some data



# 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

Connect

History **Browse**

Type here to filter the list

- Local
- Network
- Azure
  - chul3@student.unimelb.edu.au | New SQL Database... | Refresh
  - master nirserver2
  - NIRStest1 nirserver2**

Server Name: nirserver2.database.windows.net

Authentication: SQL Server Authentication

User Name: chul3@student.unimelb.edu.au

Password:

☐ Remember Password

Database Name: NIRStest1

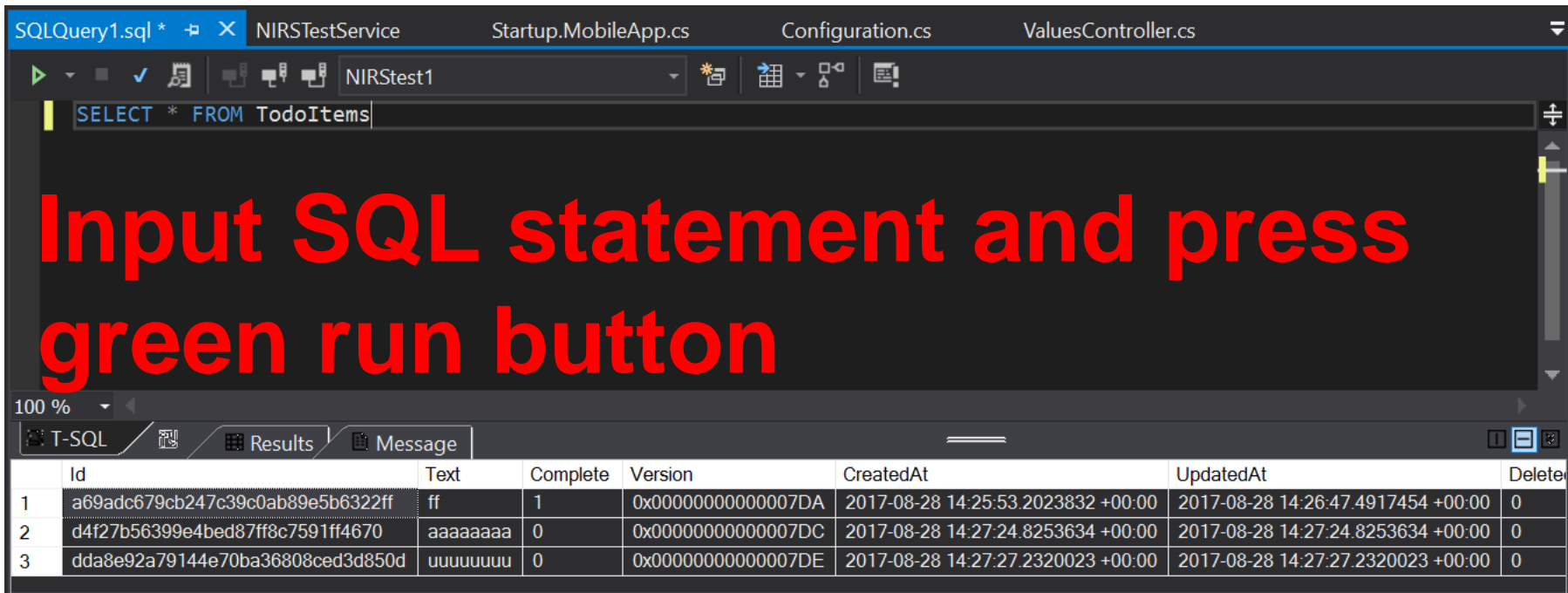
Advanced...

Connect Cancel

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



# Bonus slides: Check your SQL database using Visual Studio



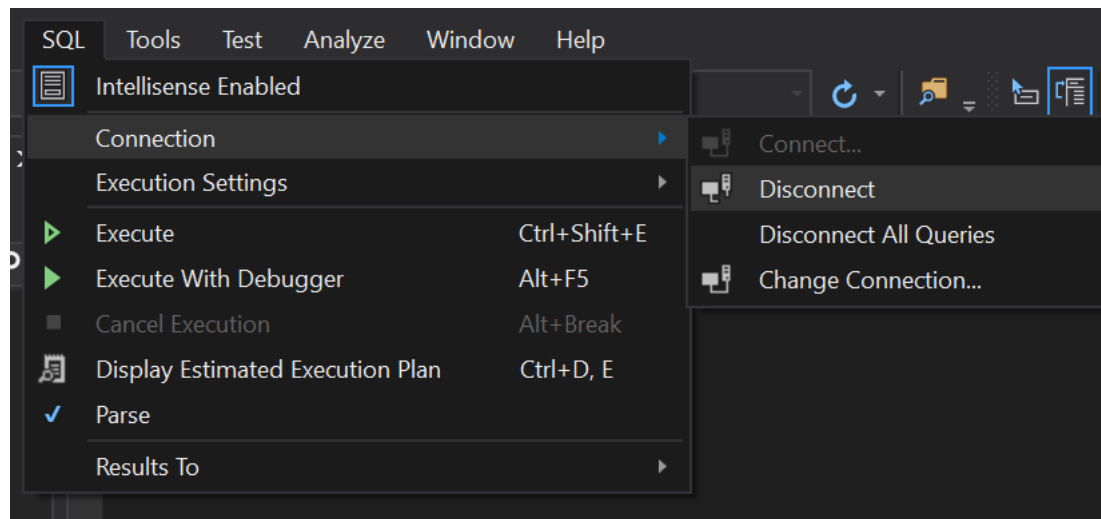
The screenshot shows the Visual Studio IDE with a solution explorer on the left containing files: SQLQuery1.sql, NIRTestService, Startup.MobileApp.cs, Configuration.cs, and ValuesController.cs. The main editor window displays the SQL query: `SELECT * FROM TodoItems`. A large red text overlay reads: "Input SQL statement and press green run button". Below the query, the "Results" tab is active, showing a table with 8 columns: Id, Text, Complete, Version, CreatedAt, UpdatedAt, and Deleted. The table contains 3 rows of data.

	Id	Text	Complete	Version	CreatedAt	UpdatedAt	Deleted
1	a69adc679cb247c39c0ab89e5b6322ff	ff	1	0x000000000000007DA	2017-08-28 14:25:53.2023832 +00:00	2017-08-28 14:26:47.4917454 +00:00	0
2	d4f27b56399e4bed87ff8c7591ff4670	aaaaaaa	0	0x000000000000007DC	2017-08-28 14:27:24.8253634 +00:00	2017-08-28 14:27:24.8253634 +00:00	0
3	dda8e92a79144e70ba36808ced3d850d	uuuuuuu	0	0x000000000000007DE	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**