# Installation Procedure – Application and Database

**NuGet Package Installation**

* Firstly we must install the necessary nuget packages by accessing the nuget packet manager at the top of your solution ‘Tools > Nuget Packet Manager > Manage Nuget Packages for Solution’ and then browsing the follwing
* Microsoft.EntityFrameworkCore
* Microsoft.EntiryFrameworkCore.SqlServer
* Microsoft.EntiryFrameworkCore.Tools
* NSwag.AspNetCore

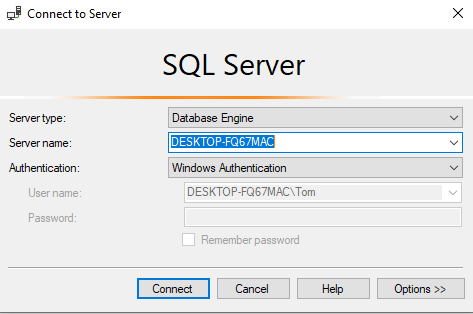
A screenshot of a computer

Description automatically generated with medium confidence

**Connecting to SQL**

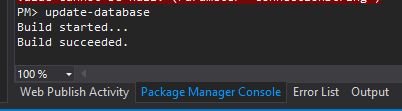
* Nextly we must connect our api to our local sql dbms





**Local Database Migration**

* Once connected to the database, go back to visual studio and open the Packet Manager Console by go to top of visual studio and selecting ‘Tools > Nuget Packet Manager > Packet Manager Console’. Once open type ‘update-database’ to push your database to your local sql server.



Graphical user interface, text, application, email

Description automatically generated

**Packaging our Application**

* Once a database has been established in your local sql server, we must now create an app service to host it on. Open your solution in visual studio, then in the menu bar at the top select “Build > Publish {ApplicationName}

A screenshot of a computer

Description automatically generated with medium confidence

**Publishing Location**

* On the publish profile screen, you can then select “Start” to begin publishing your application. You will first be prompted to select a location to publish the application, for our project we have decided to publish to Microsoft Azure.

Graphical user interface, application

Description automatically generated

**Selecting Azure Service**

* Nextly you will be prompted to select a service to host your application, for our application we have chosen the Azure App Service (Windows).

Graphical user interface, text, application

Description automatically generated

**Creating an App Service**

* Thirdly to create an azure app service, we must first login in to an Windows Azure Account, Create a unique name for our service, and then select a subscription the azure account has access to, the resource group and a hosting plan. If no resource group exists you can create a new one by clicking “New” button next to it and following the prompts; the same goes for the hosting plan.

Graphical user interface, application

Description automatically generated

**Hosting Plan Creation**

* When creating the hosting plan it is important to select a relevant size and location to the clients project.

A screenshot of a computer

Description automatically generated with medium confidence

**Completing App Service Creation**

* On the app service screen click create to begin creating your app service.

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

**Publish Settings**

* After clicking finish we’ve successfully created our app service and are almost ready to publish our API, but firstly we need to change some settings and migrate our database from our local database to the newly created app service, to modify the settings we can just click one of the blue ‘pencils’ in the summary. For our project we need to change our target runtime from ‘portable’ to ‘win-x86’.

When publishing to Azure, Visual Studio compresses the project as a package and uploads the project to your Microsoft Azure Account. Your web application will automatically launch in your browser.  
A screenshot of a computer

Description automatically generated

**Pre-Installation Tasks**

Before publishing our application there are some tasks that should be completed:

* Testing and compiling the project in Release mode
* Removing developer code (all code used for debugging).
* Setting the target runtime to your desired platform – select portable if you are unsure.
* Create a Microsoft Azure Account / Have Access to an Account
* Ensure all Client Requirements are met, communicate with the client and ensure that you are ready to launch
* Connection String Setup

**Creating an Azure SQL Database Server**

* After adjusting the settings we are ready publish our must now migrate our database to the app server, to do this we are going to go to the Azure portal and create an Sql database server by selecting ‘Create Azure SQL’.



**Selecting Deployment Options**

* When creating an Azure Sql, select ‘Database Server’, then hit create.

Graphical user interface, application

Description automatically generated

**Basics Setup**

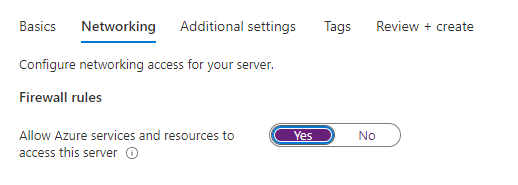
* Nextly select your subscription and resource group then create your server details, remember your administrator details as these will be needed to migrate the database and then select an apropriate location and continue

Graphical user interface, text, application, email

Description automatically generated

**Networking Setup and Deployment of our SQL Server**

* For the Networking settings, we want this this to be ‘Yes’, then hit create to create your Azure sql server.



Graphical user interface, text, application, email

Description automatically generated  
Graphical user interface, text, application, email

Description automatically generated

**Connecting to Our SQL Database Server**

* After deploying the Azure Sql server, we now want to connect to it on the dbms using its server name which we created earlier and changing the authentication to ‘SQL Server Authentication’ logging in with the administrator details created earlier (You may be prompted to log in to azure). If successful, we should be connected to both the local machine and the newly created sql server.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Database Migration to SQL Server**

* To now migrate the data from the local machine to the cloud, right click on the database and select ‘Tasks > Deploy database to Microsoft Azure Sql Database’ to open the wizard, one open click next.

Graphical user interface, text, application, email

Description automatically generated

**Creating a Database on the SQL Server**

* Nextly, connect to your server and create your databases name. For the settings we have chosen to go with this for this project, however they are subject to change

Graphical user interface, text, application

Description automatically generated

**Database Deployment on the SQL Server**

* After that’s that done, hit next then finish to import the database to azure, we are now ready to deploy!

Table

Description automatically generated

Graphical user interface, text

Description automatically generated

**API Connection to SQL Server**

* Once that’s completed we must finally connect our API to the azure sql database, we do this by first going to the newly created sql database we just imported on the Azure portal and select ‘Connection Strings’

Graphical user interface, text, application, email

Description automatically generated

**Connection String Setup**

* This will open up a page with the connection string to that Azure sql database, we must copy that connection string and paste it paste it in our api’s appsettings.json (ensuring to add our password)

Text

Description automatically generated

**Package Installation - Complete**

Once that is completed we can now finally publish our api!

Background pattern

Description automatically generated

**Deployment Testing**

* To view if our api deployed successfully, go to app service on the azure portal and copy the url and add an endpoint (eg: [https://flawlessfeedbackapi20211122063708.azurewebsites.net/{Endpoint}/{id}](https://flawlessfeedbackapi20211122063708.azurewebsites.net/%7bEndpoint%7d/%7bid%7d))



Graphical user interface, text, application, email

Description automatically generated

* Successful Deployment.

Graphical user interface, text, application

Description automatically generated

# Post-Installation Tasks

**Continued Support For the Application**

* Post installation we can provide further support by:
* Creating backups on a local database in case of data-loss or data-tampering.
* Monitor end user feedback and apply system or qol changes. Depending on how large the update, a new version may be considered.
* Continual communication with client’s, a business may want a new feature or an existing one improved.

**Creating a New Login**

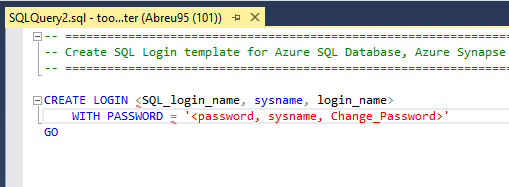
* We are now going to create a user with read/write functions, to do this first login to you Azure sql server and in the object explorer and right click the Login folder located under security and select ‘New Login…’.

A picture containing graphical user interface

Description automatically generated

**SQL Query (Login Creation)**

* This will bring up a Sql query page with a template to create a new Login, using this we can create our username and password.



Text

Description automatically generated

**Execute Query**

* Our new Login should appear after refreshing the server.



Graphical user interface, text

Description automatically generated

**Creating a New User**

* We must now create a new user to set the permissions of our user, this will once again open up a templated sql query.

Table

Description automatically generated

**SQL Query (User Creation)**

* Within this template we will create a new User for our Login, and set up its roles with ‘datawriter’ and ‘datareader’ while also removing ‘owner’. This will set up our login to only have the ability to read and write and nothing else. The owner role has complete access to the database, so we remove that.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Connecting our User to the Server**

* Login to our Sql server again using the newly created user, but firstly be sure to set the connection properties to the name of your database by going ‘Options >> Connection Properties’.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**New User Created.**

* We are now connected to the App service with our new user.

Text

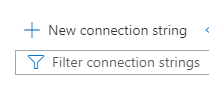
Description automatically generated with medium confidence

**Microsoft Azure Connection String Setup**

* We will now set up our connection string in azure by going to our App service > Configuration the pressing the ‘ + new connection string’

Graphical user interface, text, application, email

Description automatically generated



**Add Connection String**

* When adding the connection string, the name must be the same as our API’s connection string.

Ensure that you have added the correct log in to the connection string value and have set the type to SQLAzure. Then hit complete.

Graphical user interface, text, application

Description automatically generated

# Front-End Deployment Procedure

**Publishing Application**

* Firstly as we did with our backend, we will begin publishing our Front-End, this is done by going to ‘Build -> Publish {Application Name}.

A screenshot of a computer

Description automatically generated with medium confidence

**Publishing Application**

* Begin publishing by clicking the start button

Graphical user interface, application

Description automatically generated

**Publishing Location**

* Similarly to the back-end, we are selecting Azure as our publishing Location

A screenshot of a computer

Description automatically generated

**Publishing Service**

* Also similarly to the back-end, we are selecting an Azure app service as our publishing service

A screenshot of a computer

Description automatically generated

**Creating an App service**

* When creating an app service for the front end, we must ensure that it using a different app service, to do this we are going to create a new app service for the front end application

Graphical user interface, text, application

Description automatically generated

**App Service Settings**

* Ensure that you’ve selected the correct resource group and hosting plan then hit create  
  Graphical user interface, application

  Description automatically generated

**Finishing Up**

* Once done select finish, finally before publishing we are going to adjust the runtime settings to be win-x86  
  Graphical user interface, application

  Description automatically generated

**Target Runtime**

finally before publishing we are going to adjust the runtime settings to be win-x86

A screenshot of a computer

Description automatically generated

**Publish Application**

We can now publish our front end.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

**Pre-Installation Tasks**

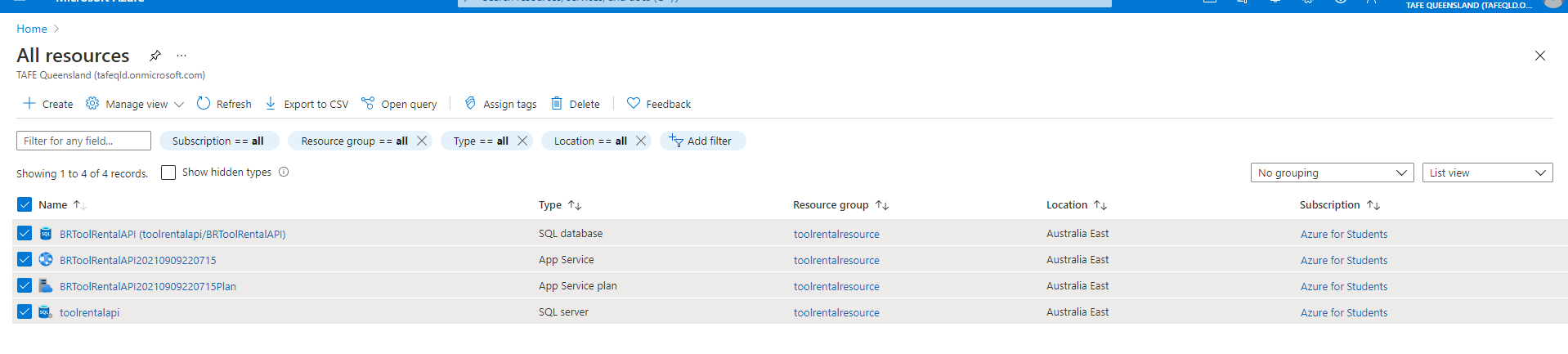
Before publishing our application there are some tasks that should be completed:

* Testing and compiling the project in Release mode
* Removing developer code (all code used for debugging).
* Setting the target runtime to your desired platform – select portable if you are unsure.
* Create a Microsoft Azure Account / Have Access to an Account
* Ensure all Client Requirements are met, communicate with the client and ensure that you are ready to launch
* Connection String Setup

# Uninstallation Procedure

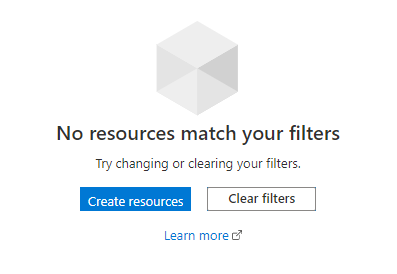
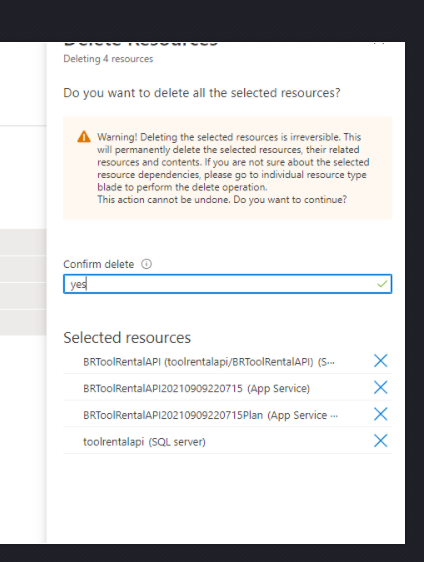
**Deleting Resources**

* To uninstall, to your all resources page and select all the resources of your resource group then select delete



**Deletion Confirmation (Resources)**

* Confirming will delete all resource groups and will be irreversible, confirm to continue,



**Removing the Resource Group**

* Lastly navigate to your resource group and delete that to fully uninstall your project off azure.

