

Module 4: Storing SQL Data in Azure

Lab: Storing Event Data in Azure SQL Databases

Scenario

Now that the web application is ready for publishing, you can begin migrating the web application to Azure by creating a database in Azure. You decided to use a database initializer and Entity Framework Code First to automate the creation of your database in SQL Database.

Objectives

After you complete this lab, you will be able to:

- Create an Azure SQL Database server and a database instance by using the Azure Portal.
- Use Entity Framework Code First to initialize and seed a database in the cloud.
- Use the Azure SQL Database Portal to view live data in the cloud.

Lab Setup

Estimated Time: 45 minutes

Before starting this lab, you must complete the lab in Module 2. For the lab in this module, you will use the available host machine. Also, you must complete the following steps:

1. On the host computer, click **Start**, type **Remote**, and then click **Remote Desktop Connection**.
2. In Remote Desktop Connection, provide the name of your virtual machine in the **Computer** box by using the following format:
 - **[Your VM IP Address]:[Your VM RDP Port]**

Note: The name and port for your virtual machine might be saved in the Computer drop-down list. If this is the case, use this value instead of typing it in manually. If you are unsure about your virtual machine's RDP port, use either of the Azure portals to find your virtual machine's endpoints. The endpoint with the name **Remote Desktop** is the correct port for RDP. This port is randomized to protect your virtual machine from unauthorized access.

3. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.
4. If necessary, sign in by using the following credentials:
 - User name: **Student**
 - Password: **AzurePa\$\$w0rd**
5. Verify that you received the credentials to sign in to the Azure portal from your training provider. You will use these credentials and the Azure account throughout the labs in this course.

Exercise 1: Creating an Azure SQL Databases Instance

Scenario

In this exercise, you will:

- Create a SQL Server instance in Azure.
- Create a database in Azure.

The main tasks for this exercise are as follows:

1. Sign in to the Azure Portal.
2. Create an Azure SQL database by using the Azure Portal.

Task 1: Sign in to the Azure Portal

1. Sign in to the Azure Portal (<https://portal.azure.com>).

Task 2: Create an Azure SQL database by using the Azure Portal

1. View the list of SQL databases for your subscription.
2. Create a SQL database by using the following details:
 - Name: **db20532**
 - Pricing Tier: **Basic**
 - Resource Group: **20532**
 - Server Name: **sv20532[Your Name Here]**
 - Server Admin Login: **testuser**
 - Server Admin Password: **TestPa\$\$w0rd**
 - Location: **Select the region that is closest to your location**
3. Record the server and database names associated with the newly created SQL Database instance.

Results: After completing this exercise, you will have created both servers and databases in the SQL Database service.

Exercise 2: Using Entity Framework with Local SQL Server

Scenario

In this exercise, you will:

- Test the web application with local SQL data.

The main tasks for this exercise are as follows:

1. Run the ASP.NET web application to view events from the local SQL database.

Task 1: Run the ASP.NET web application to view events from the local SQL database

1. In Visual Studio, open the **Contoso.Events** solution from the following location:
 - File location: **Allfiles (F):\Mod04\Labfiles\Starter\Contoso.Events**

2. Debug the **Contoso.Events.DataGeneration** console project.

Note: This Data Generation console application creates some sample data in your local SQL database (LocalDb) so that you can test the ASP.NET web application.

3. Debug the **Contoso.Events.Web** web project.

Exercise 3: Using Entity Framework with Azure SQL Databases

Scenario

In this exercise, you will:

- Configure an Entity Framework Code First **DatabasesInitializer**.
- Implement a **Seed** method for the initializer.
- Publish a Web App application.
- View tables and data by using the Azure SQL Database Portal.

The main tasks for this exercise are as follows:

2. Configure DbContext with a new DatabasesInitializer.
3. Implement seed data with DbContext.
4. Publish the cloud application with updated DbContext to Azure.
5. Verify that the Azure Web App website is using the new data.
6. Sign in to the Azure Portal.
7. View the migrated data in the Azure SQL Portal.

Task 1: Configure DbContext with a new DatabasesInitializer

1. In the **Contoso.Events.Data** project, create a new **EventsContextInitializer** class
2. Update the **EventsContextInitializer** class with the **public** accessor.
3. Update the **EventsContextInitializer** class to inherit from the **CreateDatabaseIfNotExists** generic class.
4. Add a **using** statement for the **System.Data.Entity** namespace.
5. In the **Contoso.Events.Data** project, open the **EventsContext** class
6. Update the **EventsContext** static constructor with the following lines of code:

```
Database.SetInitializer<EventsContext>(new EventsContextInitializer());
```

Task 2: Implement seed data with DbContext

1. Override the **Seed** method of the **EventsContextInitializer** class.
2. Add a **using** statement for the **Contoso.Events.Models** namespace.
3. At the end of the **Seed** method, create an instance of the **EventItem** class by using the following values:

- EventKey: **"FY17SepGeneralConference"**
 - StartTime: **DateTime.Today**
 - EndTime: **DateTime.Today.AddDays(3d)**
 - Title: **"FY17 September Technical Conference"**
 - Description: **"Sed in euismod mi."**
 - RegistrationCount: **1**
4. At the end of the **Seed** method, add the newly created **EventItem** to the **context.Events** collection.
 5. At the end of the **Seed** method, create an instance of the **Registration** class by using the following values:
 - EventKey: **"FY17SepGeneralConference"**
 - FirstName: **"Aisha"**
 - LastName: **"Witt"**
 6. At the end of the **Seed** method, add the newly created Registration to the **context.Registrations** collection.
 7. At the end of the **Seed** method, invoke the **context.SaveChanges()** method.
 8. Build the **Contoso.Events.Data** project.

Task 3: Publish the web application with updated DbContext to Azure

1. Update the **Web.Release.config** file in the **Contoso.Events.Web** project to use a connection string to your Azure SQL database.
2. Publish the **Contoso.Events.Web** project to a new Web App instance in Azure App Service by using the following details:
 - Name: **Keep the auto-generated Web App name**
 - Resource Group: **Create a New Resource Group named *TestSQL***
 - App Service Plan: **Keep the auto-generated App Service Plan name**

Task 4: Verify that the Azure Web App website is using the new data

1. View the website for your published Web App.

Task 5: Sign in to the Azure Portal

2. Sign in to the Azure Portal (<https://portal.azure.com>).

Task 6: View the data in the Azure SQL Database

1. View the list of SQL databases for your subscription.
2. View the blade for your previously created SQL database:
 - Name: **db20532**
3. Update the firewall to allow your virtual machine to remotely access the Azure SQL Database server instance.
4. Manage your SQL database from Visual Studio using the following credentials:

- Username: **testuser**
 - Password: **TestPa\$\$w0rd**
4. View the data in the **dbo.Events** table.

Results: After completing this exercise, you will have configured Entity Framework to initialize a new database with seed data.

©2016 Microsoft Corporation. All rights reserved. The text in this document is available under the [Creative Commons Attribution 3.0 License](#), additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.