

# Module 4: Storing SQL Data in Azure

## Lab: Storing Event Data in Azure SQL Databases

### Exercise 1: Creating an Azure SQL Databases Instance

#### Task 1: Sign in to the Azure Portal

1. On the Start screen, click the **Internet Explorer** tile.
2. Go to <https://portal.azure.com>.
3. In the email address box, type the email address of your Microsoft account.
4. Click **Continue**.
5. In the password box, type your password for your Microsoft account.
6. Click **Sign In**.

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

1. In the navigation pane on the left side, scroll down, and then click **More Services**.
2. In the **Browse** blade that displays, click **SQL databases**.
3. At the top-left corner of the portal, click + **New**.
4. In the **New** blade that displays, click **Databases**, and then **SQL Database** from **Databases** blade.
5. In the **SQL Database** blade that displays, locate the **Database name** box and provide the value **db20532**.
6. Click **Pricing Tier**.
7. In the **Choose your pricing tier** blade that displays, click the **View all** link and then select the **Basic** option.
8. Click **Select** to close the blade.
9. Click **Select Source**. Perform the following actions:
  - a. Select the **Blank Database** option.
  - b. In the **SQL database** blade, click **Server**.
  - c. In the **Server** blade that displays, click **Create a new server**.
  - d. In the **New Server** blade that displays, locate the **Server Name** box.
  - e. In the **Server Name** box, type **sv20532[Your Name Here]**.
  - f. In the **Server Admin Login** box, type **testuser**.
  - g. In the **Password** box, type **TestPa\$\$w0rd**.
  - h. In the **Confirm Password** box, type **TestPa\$\$w0rd**.
  - i. In the **Location** list, select the region that is closest to your location.
  - j. In the **New server** blade, click **Select**.
10. In the **SQL database** blade, locate the **Resource group** section and select the existing **20532** resource group.

11. Click **Create** to create the SQL database and server.
12. Write down the server and database names for the new 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

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

1. On the Start screen, click the **Desktop** tile.
2. On the taskbar, click the **File Explorer** icon.
3. In the *This PC* window, go to **Allfiles (F):\Mod04\Labfiles\Starter\Contoso.Events**, and then double-click **Contoso.Events.sln**.
4. In the **Solution Explorer** pane, right-click the **Contoso.Events.DataGeneration** project, point to **Debug**, and then click **Start New Instance**.

**Note:** The Data Generation script takes between one to two minutes to run.

5. In the **Solution Explorer** pane, right-click the **Contoso.Events.Web** project, and then click **Set as Startup Project**.
6. On the **Debug** menu, click **Start Debugging**.
7. On the home page of the web application, verify that it shows a list of multiple events.
8. Close the tab that is displaying the website.

## Exercise 3: Using Entity Framework with Azure SQL Databases

### Task 1: Configure DbContext with a new DatabaseInitializer

1. In the **Solution Explorer** pane, right-click the **Contoso.Events.Data** project, point to **Add** and then click **New Item**.
2. In the **Add New Item** dialog box, perform the following steps:
  - a. Expand **Installed**, expand **Visual C# Items**, and then click **Code**.
  - b. Click the **Class** template.
  - c. In the **Name** box, type **EventsContextInitializer.cs**.
  - d. Click **Add**.
3. Add the following **using** statement at the top of the code file:

using System.Data.Entity;

4. In the **EventsContextInitializer** class, add the **public** accessor to the left of the class definition:

```
class EventsContextInitializer
```

5. Verify that the updated class definition looks like the following line of code:

```
public class EventsContextInitializer
```

6. In the **EventsContextInitializer** class, add the **CreateDatabaseIfNotExists<EventsContext>** inheritance statement to the right of the class definition:

```
public class EventsContextInitializer
```

7. Verify that the updated class definition looks like the following line of code:

```
public class EventsContextInitializer : CreateDatabaseIfNotExists<EventsContext>
```

8. In the Solution Explorer pane, expand the **Contoso.Events.Data** project.

9. In the **Contoso.Events.Data** project, open the **EventsContext.cs** file.

10. Within the static constructor **static EventsContext()**, add the following line of code:

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

11. Save the *EventsContext.cs* file.

## Task 2: Implement seed data with DbContext

1. In the Solution Explorer pane, expand the **Contoso.Events.Data** project.

2. In the **Contoso.Events.Data** project, open the **EventsContextInitializer.cs** file.

3. Add the following method declaration to the **EventsContextInitializer** class:

```
protected override void Seed(EventsContext context){ }
```

4. Add the following **using** statements at the top of the code file:

```
using Contoso.Events.Models;
```

5. Place the cursor between the opening and closing curly brackets **{ }** to the right of the **Seed(EventsContext context)** method, and then type the following lines of code:

```
if (context.Events.Count() == 0)
{
    Event eventItem = new Event();
    eventItem.EventKey = "FY17SepGeneralConference";
    eventItem.StartTime = DateTime.Today;
    eventItem.EndTime = DateTime.Today.AddDays(3d);
    eventItem.Title = "FY17 September Technical Conference";
    eventItem.Description = "Seed in euismod mi.";
    eventItem.RegistrationCount = 1;
    context.Events.Add(eventItem);
}
```

```
if (context.Registrations.Count() == 0)
{
    Registration registrationItem = new Registration();
```

```

        registrationItem.EventKey = "FY17SepGeneralConference";
        registrationItem.FirstName = "Aisha";
        registrationItem.LastName = "Witt";
        context.Registrations.Add(registrationItem);
    }

```

```
context.SaveChanges();
```

6. Save the *EventsContextInitializer.cs* file.
7. In the Solution Explorer pane, right-click the **Contoso.Events.Data** project, and then click **Build**.

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

1. In the **Solution Explorer** pane, expand the **Contoso.Events.Web** project.
2. In the **Solution Explorer** pane, expand the **Web.config** file in the **Contoso.Events.Web** project.
3. Double-click the **Web.Release.config** file.
4. In the **Web.Release.config** file, update the connection string using the key **EventsContextConnectionString** with the following values:
  - [database]: **db20532**
  - [login]: **testuser**
  - [server]: **sv20532[Your Name Here]**. (Note that there are two different places to replace [server].)
  - [password]: **TestPa\$\$w0rd**

**Note:** Ensure that you remove the square brackets as you replace each placeholder.

5. Save the **Web.Release.config** file.
6. In the **Solution Explorer** pane, right-click the **Contoso.Events.Web** project, and then click **Publish**.
7. In the Publish Web window, click **Microsoft Azure App Service**.
8. In the **App Service** dialog, perform the following steps:
  - a. Select your Azure subscription.
  - b. Ensure that the **Resource Group** option is selected in the **View** list.
  - c. Click the **New** button.
9. In the **Create App Service** dialog, perform the following steps:
  - a. Ensure that you have an auto-generated Web App name. If not, enter a globally unique name.
  - b. Select your Azure subscription.
  - c. Click the **New** button immediately to the right of the **Resource Group** dialog box.
  - d. In the **Resource Group** dialog box, provide the value **TestSQL**.

- e. Click the **New** button immediately to the right of the **App Service Plan** dialog box.
- f. Ensure that you have an auto-generated App Service Plan name. If not, provide the value **TestSQLPlan**.
- g. In the **Location** list, select the region that is closest to your location. **Ensure** that this region is same as the **region\*** where you created SQL Database.
- h. In the **Size** list, select the **Free** option.
- i. Click the **OK** button to close the **Configure App Service Plan** dialog.
- j. Click the **Create** button to create your App Service instance.

**Note:** The deployment process for the new App Service is relatively short and should take 2-5 minutes.

- 10. In the **Publish Web** dialog box, perform the following steps:
  - a. Leave the default values in all the fields.
- 11. Click **Publish**.

**Note:** It typically takes five to ten minutes for the publish process to complete. You can track the progress of your publish in the Microsoft Azure Activity Log (**View > Other Windows > Microsoft Azure Activity Log**) pane that displays when you publish your Web App project.

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

- 1. Wait for the publish process to complete and the console window to display the message **Complete**.

**Note:** The publish process is complete when the message “**Complete**” displays in the **Microsoft Azure Activity Log**’s history console. The green circular indicator in the Activity Log does not indicate that the publish process is complete, but it indicates that the package is uploaded successfully.

- 2. In the **Azure App Service Activity** pane, click the hyperlink that directs you to the published web application.
- 3. Verify that the website displays the single event that you created in your Entity Framework context initializer.

#### Task 5: Sign in to the Azure Portal

- 1. On the Start screen, click the **Internet Explorer** tile.

2. Go to *https://portal.azure.com*.
3. In the email address box, type the email address of your Microsoft account
4. In the password box, type the password of your Microsoft account, and then click **Sign In**.

#### **Task 6: View the data in the Azure SQL Database**

1. In the navigation pane on the left side, scroll down, and then click **More Services**.
2. In the **Browse** blade that displays, click **SQL databases**.
3. In the list of **SQL Databases**, select the SQL database named **db20532**.
4. In the **db20532 - SQL database** blade, locate the **Essentials** panel.
5. Locate the **Server name** section and click on the associated hyperlink to navigate to the server blade.
6. In the **SQL server** blade, locate the **Essentials** panel.
7. Locate the **Firewall** section and click on the associated hyperlink to navigate to manage firewall settings.
8. In the **Firewall settings** blade, click the **Add client IP** button to add your virtual machine's IP Address to the list of allowed IP Address ranges.
9. Click on **Save** at the top of the blade. Once saved, close the confirmation dialog by clicking the **Ok** button.

**Note:** It might take couple of minutes for the firewall changes to get updated on server.

10. Return to the open instance of **Visual Studio**.
11. In Visual Studio, open the **View** menu and then select the **Server Explorer** option.
12. **Expand** the **Data Connections** node.
13. Right click on **Data Connections** and click on **Add Connection**.
14. Choose **Microsoft SQL Server** for data source and click **Continue**.
15. In the **Add Connection** wizard, provide following values and click **OK**.
  - a. In the **server name** box, type **sv20532[your name].database.windows.net**.
  - b. Select **Use SQL Server Authentication**.
  - c. In the **Username** box, type **testuser**.
  - d. In the **Password** box, type **TestPa\$\$w0rd**.
  - e. In the **Select or enter a database name** dropdown, select **db20532**.
  - f. Click the **OK** button.

**Note:** If firewall rules are not updated on the server, you may have to wait a few more minutes before proceeding.

16. On Visual Studio **Server Explorer**, expand **Data Connections** then **sv20532[your name].db20532.dbo** and then the **Tables** node.
17. Right click **Tables** and click **Refresh**.
18. Right click **Events** table, and then select **Show Table Data**.
19. In the **dbo.Events** table, view the single record.
20. Close the **Events** table window.
21. Close the **Internet Explorer** application.
22. Close the **Contoso.Events – Microsoft Visual Studio** window.

**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.