# Module 10: Securing Azure Web Applications

# Lab: Integrating Azure Active Directory with the Events Administration Portal

### Scenario

Even though the Contoso Events web application is public, the Administration application should be locked down to users only from your domain. You have decided to use Azure AD and ASP.NET identity to provide this functionality. In this lab, you will create a new ASP.NET project by using the ASP.NET identity framework and integrate the project with Azure AD. The website will then use your organization accounts for signing in.

### Objectives

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

* Create an Azure AD directory by using the Classic Portal.
* Create users in Azure AD by using the Classic Portal.
* Create a new MVC project that uses Azure AD organizational accounts for security.

### Lab Setup

**Estimated Time: 60 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.

1. In Remote Desktop Connection, click **Connect**. Wait until the RDP client accesses the virtual machine.
2. If necessary, sign in by using the following credentials:
   * User name: **Student**
   * Password: **AzurePa$$w0rd**
3. Verify that you received the credentials to sign in to the Azure Classic 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 AD

### Scenario

In this exercise, you will:

* Create an Azure AD by using the Classic Portal.
* Create a Global Administrator role in the new directory.
* Create a standard User role in the new directory.

The main tasks for this exercise are as follows:

1. Sign in to the Azure Classic Portal.
2. Create a directory by using the Classic Portal.
3. Create a Global Administrator role in the directory.
4. Create a User role in the directory.

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

**Note:** For this lab, you will use the Classic Portal because the functionality to create new Azure Active Directory domains is not available in the Preview Classic Portal.Sign in to the Classic Azure Portal ([*https://manage.windowsazure.com*](https://manage.windowsazure.com)).

#### Task 2: Create an directory by using the Classic Portal

1. View the list of Active Directory directories for your subscription.
2. Create a new directory by using the following details:
   * Name: **20532**
   * Domain Name: **ad20532[*Your Name*]**
   * Country or Region: **Select your current location**

#### Task 3: Create a Global Administrator role in the directory

1. Go to your newly created directory’s Users list.
2. Add a new user to your directory by using the following details:
   * Type of User: **New user in your organization**
   * User Name: **admin**
   * Domain: **ad20532[*Your Name*]**
   * First Name: **Admin**
   * Last Name: **User**
   * Display Name: **Admin User**
   * Role: **Global Admin**
   * Alternate E-mail Address: [***example@contoso.com***](mailto:example@contoso.com)

* **Note:** Make sure you record the temporary password that is generated when you create the new user. In case you forget, you can always reset the user’s password by using the **Reset Password** button that is next to the **Add User** button in the Classic Portal.

#### Task 4: Create a User role in the directory

1. Add a new user to your directory by using the following details:
   * Type of User: **New user in your organization**
   * User Name: **standard**
   * Domain: **ad20532[*Your Name*]**
   * First Name: **Standard**
   * Last Name: **User**
   * Display Name: **Standard User**
   * Role: **User**

* **Note:** Make sure you record the temporary password that is generated when you create the new user. In case you forget, you can always reset the user’s password by using the **Reset Password** button that is next to the **Add User** button in the Classic Portal.

**Results:** After completing this exercise, you will have created an Azure AD and populated the directory with users.

## Exercise 2: Securing an Existing ASP.NET Web Application

### Scenario

In this exercise, you will:

* Use the MVC **AuthorizeAttribute** to secure a web application.
* The main tasks for this exercise are as follows:

1. Add the Authorize attribute to the *HomeController* class.
2. Debug the web application.

#### Task 1: Add the Authorize attribute to the HomeController class

1. Open the **Contoso.Events** solution from the following location:
   * File location: **Allfiles (F):\Mod10\Labfiles\Starter\Contoso.Events**
   * Open the **HomeController.cs** file in the **Contoso.Events.Management.Old** project.
   * Add the **Authorize** attribute to the **HomeController** class.

#### Task 2: Debug the web application

1. Debug the solution, and then verify that you receive a 401 (Unauthorized) error message.
2. Stop debugging.

**Results:** After completing this exercise, you will have used MVC to ensure that a user is signed in before accessing a controller or action.

## Exercise 3: Integrating Azure AD with ASP.NET Identity

### Scenario

In this exercise, you will:

* Create a new ASP.NET MVC application by using ASP.NET Identity for authentication and authorization.
* Integrate the new Website with Azure AD organization accounts.
* The main tasks for this exercise are as follows:

1. Create a new management web application by using Azure AD as the identity provider.
2. Verify that the Management web application requires a sign-in by using an organizational account.

#### Task 1: Create a new management web application by using Azure AD as the identity provider

1. In the **Administration** solution folder, create a new **Contoso.Events.Management** ASP.NET Web Application project by using the following details:
   * Project Name: **Contoso.Events.Management**
   * Project Location: **Allfiles (F):\Mod10\Labfiles\Starter\Contoso.Events**
   * Template: **MVC**
   * Authentication: **Organizational Accounts**
   * Domain: **ad20532[*Your Name*].onmicrosoft.com**
   * Account used for authentication: **admin@ad20532[*Your Name*].onmicrosoft.com**

* **Note:** When associating your new ASP.NET Web Application project with an Azure AD domain, you must sign in by using an account that has a Global Administrator role.

1. Set the **Contoso.Events.Management** project as the startup project.
2. In the **Contoso.Events.Management** project, add references to the **Contoso.Events.Models** and **Contoso.Events.ViewModels** projects.
3. Copy all the folders from the **Contoso.Events.Management.Old** project to the **Contoso.Events.Management** project.

* **Note:** Ensure that you copy the folders and not any of the individual files. Ignore the following files:
  + favicon.ico
  + Global.asax
  + packages.config
  + Web.config

1. Copy the **connectionStrings** from the web.config file in the *Contoso.Events.Management.Old* project and paste them in the new project's web.config file.

#### Task 2: Verify that the Management web application requires a sign-in by using an organizational account

1. Debug the solution.
   * Configure IIS Express to trust the self-signed certificate.
   * Verify that you are required to sign in before accessing the **Contoso.Events.Management** web application.

**Results:** After completing this exercise, you will have used an Azure AD domain with the ASP.NET Identity framework.

©2016 Microsoft Corporation. All rights reserved. The text in this document is available under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/legalcode), 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.