App Service - API Apps

- Introduction
- Creating an API App
- Deploying an API App
- Consuming an API App
- Authentication and Authorization in API App.
- Using API App in Logic App

Introduction

RESTful Service = HTTP (Protocol) + JSON / XML (Data Format)

Http Protocol Request

1) URL = http://www.demo.com/employee

2) Request method = GET / POST / PUT / MERGE / DELETE …

3) Request Body = XML or JSON

Example

- **Get All Employees:** http://www.demo.com/employee + GET
- **Get one Employee:** http://www.demo.com/employee?Id=1 + GET
- **Add an Employee:** http://www.demo.com/employee + POST + {"empName":"Sandeep",.....}
- **Update an Employee:** http://www.demo.com/employee/1 + PUT + {"empName":"Sandeep",.....}
- **Delete an Employee:** http://www.demo.com/employee/1 + DELETE

API apps in Azure App Service offer features that make it easier to develop, host, and consume APIs in the cloud and on-premises.

Web API => API App

Here are some key features of API Apps:

- **Bring your APIs as-is:** Your existing APIs can take advantage of the features of the API Apps feature of Azure App Service—**with no code changes**. Use ASP.NET, Java, PHP, Node.js or Python to build new API apps.
- **Easy consumption** - Integrated support for **Swagger API metadata** makes your APIs easily consumable by a variety of clients. Automatically generate client code for your APIs in a variety of languages including C#, Java, and Javascript

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

- **Simple access control:** With no changes to your code, you can expose APIs to only the other apps **inside of your organization** or to the **public**. Secure your API apps with identity providers **Azure Active Directory, Google, Microsoft, Facebook and Twitter**.
- **Automatic SDK generation:** API Apps lets you take your new and existing APIs and automatically generate SDKs for a variety of languages, including C#, Java, JavaScript and more—empowering your business to easily use your APIs for web, mobile and desktop applications.
- **Integration with Logic Apps** - API apps that you create can be consumed by App Service Logic Apps.

Creating an API App

You'll learn:

- How to create and deploy API Apps in Azure App Service by using tools built into Visual Studio 2017.
- How to automate API discovery by using the Swashbuckle – Swagger for WebAPI NuGet package to dynamically generate Swagger API metadata.
- How to use Swagger API metadata to automatically generate client code for an API app.

1. **Visual Studio → Create a New Project:** New →Project → ASP.NET.Core Web Application → **Choose an Azure API App**.
2. Go to Tools → NuGet Package Manager → Manage Nuget Packages for Solution → Add reference to **EntityFramework.Core**
3. Under Models folder add the following Employee and Context classes.

```
public class Employee
{
   [Key]
   public int EmpId { get; set; }
   public string Name { get; set; }
   public decimal Salary { get; set; }
}
```

4. Build the project
5. Right click on Controller Folder → Add → Controller. In the scaffolding option, select Web API Controller with actions, using Entity Framework.
6. Model Class = Employee, Data Context Class → Click on "+" and accept the default.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

Note: EmployeeController would be added to the project. Essentially, we have created **Web API** to perform CRUD operations on the Employee table.

7. Edit **Program.cs** as below

```
public static void Main(string[] args)
{
   var host = CreateHostBuilder(args).Build();

   using (var scope = host.Services.CreateScope())
   {
      var services = scope.ServiceProvider;
      try
      {
         var context = services.GetRequiredService<WebAPIDemoinCoreContext>();
         context.Database.EnsureCreated();
      }
      catch (Exception ex)
      {
         var logger = services.GetRequiredService<ILogger<Program>>();
         logger.LogError(ex, "An error occurred creating the DB.");
      }
   }
   host.Run();
}
```

About Swagger:

Swagger is a **language-agnostic** specification for **describing REST APIs**. The Swagger project was donated to the **OpenAPI** Initiative, where it's now referred to as Open API. Both names are used interchangeably; however, Open API is preferred. It allows **both computers and humans** to understand the capabilities of a service without any direct access to the implementation (source code, network access, documentation). One goal is to minimize the amount of work needed to connect disassociated services. Another goal is to reduce the amount of time needed to accurately document a service.

https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger

- Using **Swashbuckle** to Swagger enable your ASP.NET WebAPI 2 REST projects which in turn provides both human and machine readable documentation automatically, much like WSDL does for SOAP services. **Swagger is standard for documenting REST APIs**.
- **Swagger UI** allows anyone — be it your development team or your end consumers — to visualize and interact with the API's resources without having any of the implementation logic in place. It's automatically generated from your Swagger specification, with the visual documentation making it easy for backend implementation and client side consumption.

8. Add a reference to **Swashbuckle.AspNetCore**

```
services.AddSwaggerGen(options =>
{
   options.SwaggerDoc("v1", new Microsoft.OpenApi.Models.OpenApiInfo()   {
      Title = "HTTP Web API Demo",
      Version = "v1",
      Description = "The is a demo of how to use Swagger in Web API",
   });
});
```

9. Add the following to Configure Method

```
app.UseSwagger().UseSwaggerUI(c => {
 c.SwaggerEndpoint("/swagger/v1/swagger.json", "My API V1");
});
```

10. In EmployeesController.cs → Edit Post Method and add the following attribute

```
[Swashbuckle.AspNetCore.Annotations.SwaggerResponse((int)HttpStatusCode.Created)]
public IHttpActionResult PostEmployee(Employee employee)
{ . . . }
```

11. Navigate to http://localhost:1234/**swagger/docs/v1**, open the document and view in VS.NET
12. Navigate to http://localhost:1234/swagger and you will find Swagger UI has been enabled on the API.

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

13. Using the Swagger UI, we can test the API created. For example, let us go ahead and try to create a new Employee. Click on the POST option, create the employee data to be inserted in JSON format, and click on Try it out.

Deploy API App to Azure

14. **Creating an Azure SQL Database Instance by using the Azure Portal**
    - **a.** Create New SQL Server
        - **i.** Browse → SQL Servers → Add
        - **ii.** Provide unique server name (all lowercase), Server Admin login, password and other details…
        - **iii.** Create
    - **b.** Create New SQL Database
        - **i.** Browse → SQL Database → Add
        - **ii.** Select Server created in earlier step and other details
        - **iii.** Create
15. Azure Portal → Create a new API App.
16. In Visual Studio, right-click the project in **Solution Explorer** and select **Publish** from the context menu
17. Under Profile Tab → Click on Microsoft Azure App Service and Login with your credentials
18. Click on New Button,
    a. Under Hosting Change Type = "API App"
    b. Go to Services Tab → Click on SQL Database +
19. Select Existing Database Server or Create a New One.
20. Provide a Database Name, Administrator Username and Password → OK
21. Publish the Web App and Note the progress in Azure App Service Activity Window
22. Navigate to http://AppName.azurewebsites.net/swagger and you will find Swagger UI has been enabled on the API.
23. Using the Swagger UI, we can test the API created. For example, let us go ahead and try to create a new Employee. Click on the POST option, create the employee data to be inserted in JSON format, and click on Try it out.

Metadata URL in portal:

24. API App → Settings → **API Definition** → Set API definition URL = http://APIAppName.azurewebsites.net**/swagger/docs/v1**"

Developing Client Application

25. Create a New ASP.NET Console Based Application
26. Right click on the project and from the context menu select **Add → New Rest API Client → Generate with Open API**.
    a. Specification URL = http://APIAppName.azurewebsites.net**/swagger/v1/swagger.json** → **OK**
    b. This add swagger.json an swagger.cs files to the project.

Note: Now we have created the client side code from the Azure API App.

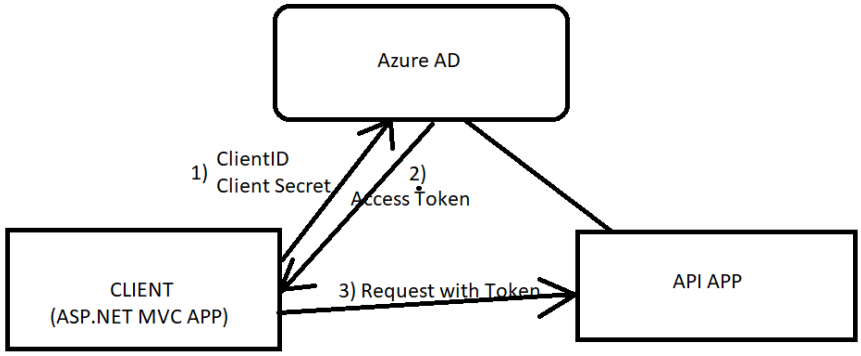
27. Edit Main as below

```
using Org.OpenAPITools.Api;
using Org.OpenAPITools.Client;
using System;
 class Program
{
   static void Main(string[] args)
   {
      var config = new Configuration();
      config.BasePath = "https://dssmyapidemoapp.azurewebsites.net/";
      EmployeesApi api = new EmployeesApi(config);
      var emps = api.EmployeesGet();
      foreach (var emp in emps)
      {
         Console.WriteLine(emp.EmpId + " " + emp.Name + " " + emp.Salary);
      }
   }
}
```

Authentication and Authorization

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)



Following Example Demostrate Authentication:

Enable Authentication for API App

1. Api App → Settings → **Authentication / Authorization**
2. App Service Authencaiton = On, Action to take when request is not authenticated = **Log in with Azure Active Directory**
3. Under Authentication Providers → Click Azure Active Directory → Management mode = **Express** → OK
4. Save
5. Api App → Settings → **Authentication / Authorization** → Under Authentication Providers → **Log in with Azure Active Directory → Manage Application Button**
    a. → Settings → Properties → Copy **Application ID** to be used later.
    b. → Settings → Keys → **Generage and Copy the App Secret**

6. Add the following to the Client Project

```
using Microsoft.IdentityModel.Clients.ActiveDirectory;
using Org.OpenAPITools.Api;
using Org.OpenAPITools.Client;
using System;

class Program
{
   static string appId = "457ffee2-d5f2-46a9-a26a-fd146b9952e6";
   static string secret = "ZRFmm_57E.g_64CTxd.XO921l.59x52rSn";
   static string tenantId = "82d8af3b-d3f9-465c-b724-0fb186cc28c7";
```


Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

```
    static void Main(string[] args)
    {
        var context = new AuthenticationContext("https://login.windows.net/" + tenantId);
        var credential = new ClientCredential(clientId: appId, clientSecret: secret);
        AuthenticationResult result = context.AcquireTokenAsync(appId, credential).Result;
        if (result == null)
            throw new InvalidOperationException("Failed to obtain the JWT token");
        var token = result.AccessToken;



        //var apiClient = new ApiClient("https://dssmyapidemoapp.azurewebsites.net/");
        //apiClient, null, null, null, token
        var config = new Configuration();
        config.DefaultHeaders.Add("Authorization", "Bearer " + token);
        config.BasePath = "https://dssmyapidemoapp.azurewebsites.net/";
        EmployeesApi api = new EmployeesApi(config);
        var emps = api.EmployeesGet();
        foreach (var emp in emps)
        {
            Console.WriteLine(emp.EmpId + " " + emp.Name + " " + emp.Salary);
        }
    }
}
```

7. Run the client application.

Using API App in Logic Apps

1. API App → Settings → CORS → Allowed Origins=* → Save
2. API App → Settings → API definition → Copy URL
3. Create a New Logic App
4. Add Triggers and Actions as per the requirement
5. To add an API App, Under Microsoft Managed API, **Search HTTP + Swagger**

6. For Swagger endpoint url = URL Copied in step2 → Select the appropriate method and Continue…

Deccansoft Software Services H.No: 153, A/4, Balamrai, Rasoolpura, Secunderabad-500003 TELANGANA, NDIA.
http://www.deccansoft.com | http://www.bestdotnettraining.com

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)