**https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1**

**Using Web API 2 with Entity Framework 6**

This tutorial teaches you the basics of creating a web application with an ASP.NET Web API back end. The tutorial uses Entity Framework 6 for the data layer, and Knockout.js for the client-side JavaScript application. The tutorial also shows how to deploy the app to Azure App Service Web Apps.

**Software versions used in the tutorial**

* Web API 2.1
* Visual Studio 2017 (download Visual Studio 2017 [here](https://visualstudio.microsoft.com/downloads/?utm_medium=microsoft&utm_source=docs.microsoft.com&utm_campaign=button+cta&utm_content=download+vs2017))
* Entity Framework 6
* .NET 4.7
* [Knockout.js](http://knockoutjs.com/) 3.1

This tutorial uses ASP.NET Web API 2 with Entity Framework 6 to create a web application that manipulates a back-end database. Here is a screen shot of the application that you will create.

[Graphical user interface, website

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image1.png)

The app uses a single-page application (SPA) design. "Single-page application" is the general term for a web application that loads a single HTML page and then updates the page dynamically, instead of loading new pages. After the initial page load, the app talks with the server through AJAX requests. The AJAX requests return JSON data, which the app uses to update the UI.

AJAX isn't new, but today there are JavaScript frameworks that make it easier to build and maintain a large sophisticated SPA application. This tutorial uses [Knockout.js](http://knockoutjs.com/), but you can use any JavaScript client framework.

Here are the main building blocks for this app:

* ASP.NET MVC creates the HTML page.
* ASP.NET Web API handles the AJAX requests and returns JSON data.
* Knockout.js data-binds the HTML elements to the JSON data.
* Entity Framework talks to the database.

**See this app running on Azure**

Would you like to see the finished site running as a live web app? You can deploy a complete version of the app to your Azure account by selecting the following button.

[](https://azuredeploy.net/?WT.mc_id=deploy_azure_aspnet&repository=https://github.com/tfitzmac/BookService)

You need an Azure account to deploy this solution to Azure. If you do not already have an account, you have the following options:

* [Open an Azure account for free](https://azure.microsoft.com/pricing/free-trial/?WT.mc_id=A443DD604) - You get credits you can use to try out paid Azure services, and even after they're used up you can keep the account and use free Azure services.
* [Activate MSDN subscriber benefits](https://azure.microsoft.com/pricing/member-offers/msdn-benefits-details/?WT.mc_id=A443DD604) - Your MSDN subscription gives you credits every month that you can use for paid Azure services.

**Create the project**

Open Visual Studio. From the **File** menu, select **New**, then select **Project**. (Or select **New Project** on the Start page.)

In the **New Project** dialog, select **Web** in the left pane and **ASP.NET Web Application (.NET Framework)** in the middle pane. Name the project **BookService** and select **OK**.

[Graphical user interface, text, application, email

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image11.png)

In the **New ASP.NET Project** dialog, select the **Web API** template.

[Graphical user interface, application

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image12.png)

Select **OK** to create the project.

**Configure Azure settings (optional)**

After you create the project, you can choose to deploy to Azure App Service Web Apps at any time.

1. In Solution Explorer, right-click on your project and select **Publish**.
2. In the window that appears, select **Start**. The **Pick a publish target** dialog box appears.

[Graphical user interface, text, application

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image14.png)

1. Select **Create Profile**. The **Create App Service** dialog box appears.

[Graphical user interface, application

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image15.png)

Accept the defaults, or enter different values for the application name, resource group, hosting plan, Azure subscription, and geographical region.

1. Select **Create a SQL database**. The **Configure SQL Server** dialog box appears.

[Table

Description automatically generated with low confidence](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image16.png)

Accept the defaults or enter different values. Enter an **Administrator Username** and **Administrator Password** for your new database. Select **OK** when you're done. The **Create App Service** page reappears.

1. Select **Create** to create your profile. A message appears in the lower-right corner indicating that deployment is in progress. After a short while, the **Publish** window reappears.

[A screenshot of a computer

Description automatically generated](https://docs.microsoft.com/en-us/aspnet/web-api/overview/data/using-web-api-with-entity-framework/part-1/_static/image17.png)

The profile you created to deploy the app is now available.