## **Objectives**

* Explain about ASP.NET Web API Get method
* Explain how EF Database First Approach and ASP.NET Web API can be used to read data which is stored in the SQL Server

In this hands-on lab, you will learn how to:

* Implement an ASP.NET Web API service that reads data that is stored in the SQL Server using EF6 Database First Approach
* Implement the GET method in ASP.NET Web API
* Use Postman to issue Get request to the Web API service

**Prerequisites**

The following is required to complete this hands-on lab:

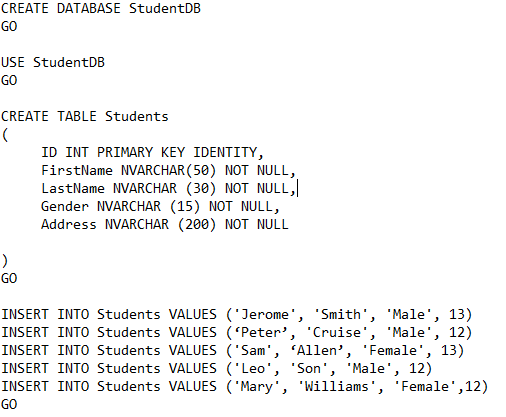
* Visual Studio 2017 Community Edition
* Web API 2.x
* SQL Server 2014
* Postman

## **Notes**

Estimated time to complete this lab: **90 minutes.**

**Task 1 – Create database table to perform read operation**

1. Create a database called **StudentDB** in SQL Server
2. Creates the Students table and populate it with some test data as shown in the below image.

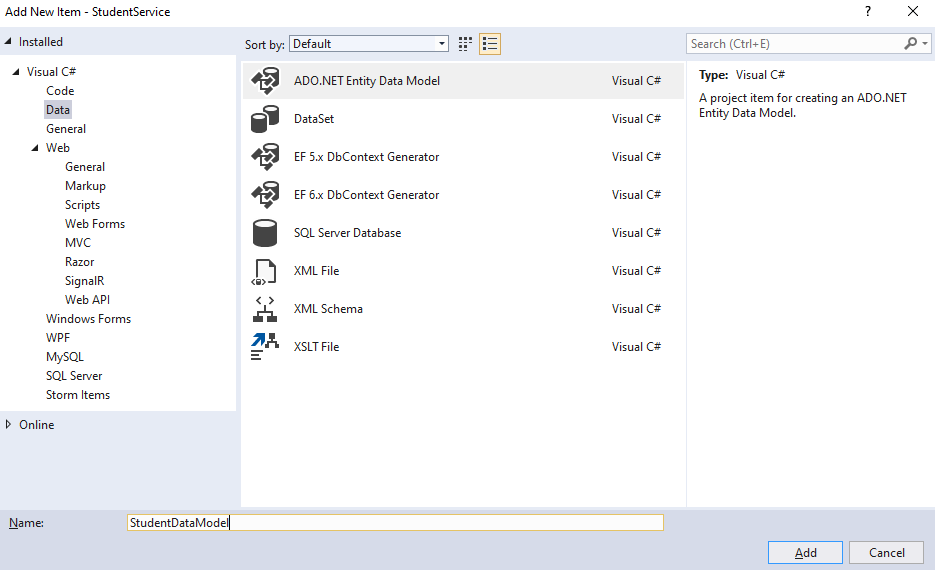


**Task 2-Creating a new ASP.NET Web API Project**

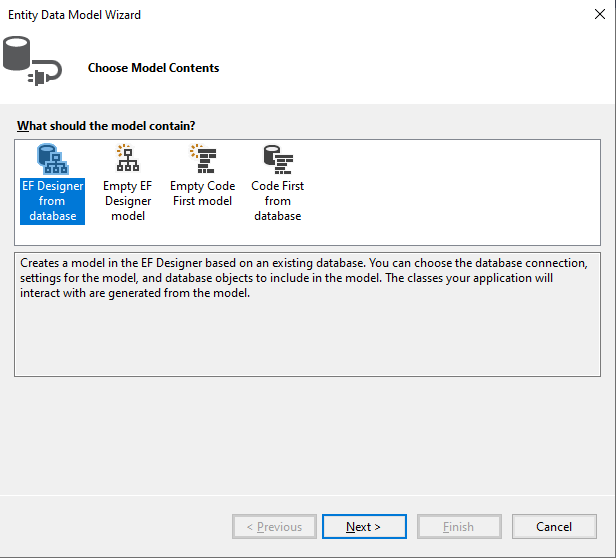
1. Refer previous exercises for creating a new ASP.NET Web API project.
2. Name your Web API project **StudentService**

**Task 3-Adding ADO.NET Entity Data Model to retrieve data from database**

1. Right-click on the Models folder and then select **Add – New Item** option which will open the Add New Item window and from the **“Add New Item”** window select the **“Data”** option from the left pane and from the middle pane select **ADO.NET Entity Data Model**. In the Name text box, type **StudentDataModel** and finally click the **Add** button as shown in the below image.



1. Once you click on the **Add** button, it will open the **Entity Data Model Wizard** and from that wizard select “**EF Designer from database”** option and click on the “**Next”** button as shown below.



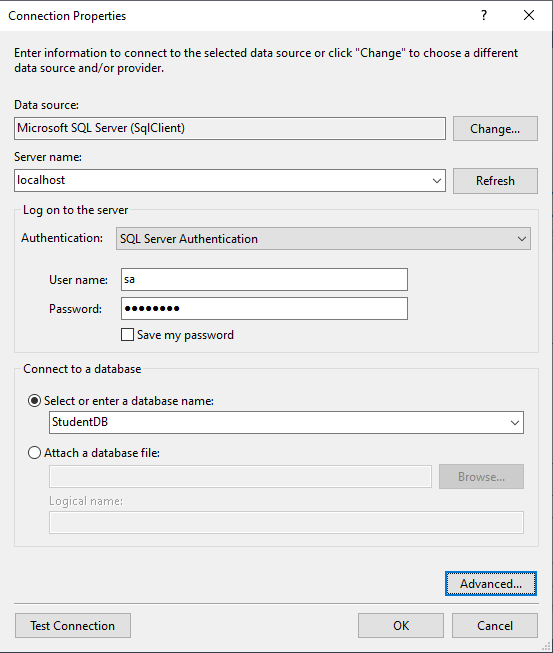
1. On the next screen, click on the “**New Connection**” button
2. Once you click on the **New Connection** Button it will open the **Connection Properties** window. On “**Connection Properties**” window, set

**Server Name =** <your\_server\_name>

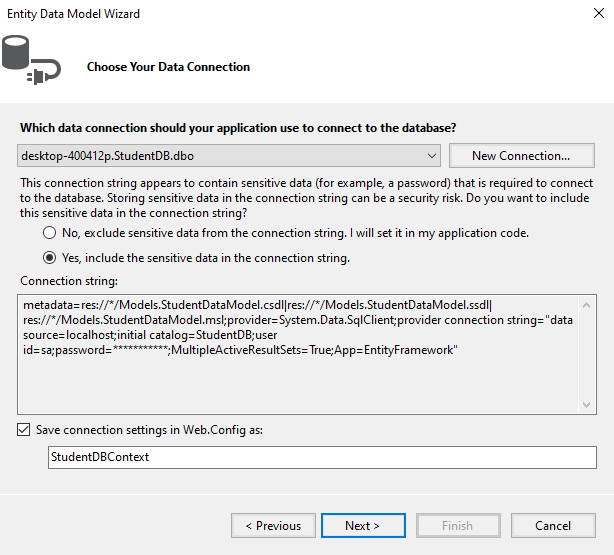
**Authentication =** Select the authentication type

**Select or enter a database name =** StudentDB

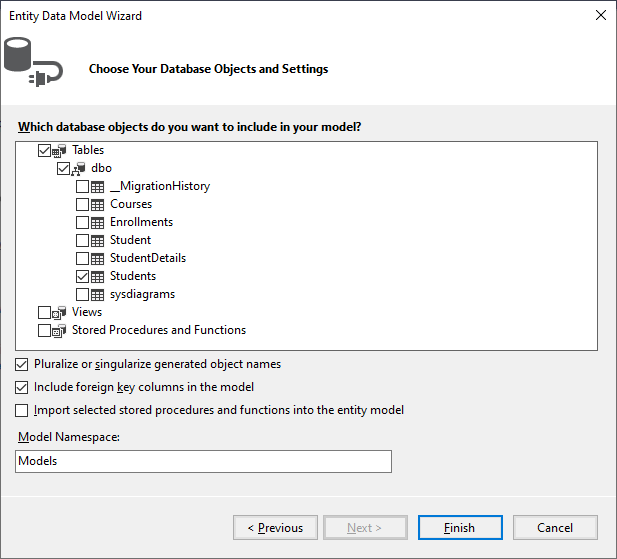
Click the **OK** button as shown below.



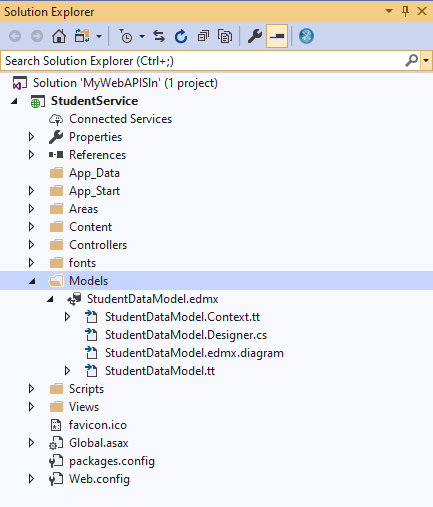
1. Once you click on the **OK** button it will navigate back to the Choose Your Data Connection wizard. Here Modify the Connection String as **StudentDBContext** and click on the Next Button as shown in the below image.



1. On the next screen, make sure that “Entity Framework 6.x” is selected.
2. On Choose Your Database Objects and Settings screen, select the **“Students”** table, provide the model namespace name and click on Finish button as shown below.



1. Once you click on the Finish Button the following edmx file will be generated within the Models folder as shown below.

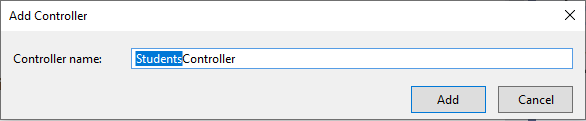


##### Task 4 - ****Adding Web API Controller****

##### Right-click on the Controllers folder and select****Add – Controller****option and then select “****Web API 2 Controller – Empty****” and click on the “****Add****” button as shown below.

##### 

1. On the next screen set, the Controller Name as **StudentsController** and click on the **Add** button as shown in the below image.



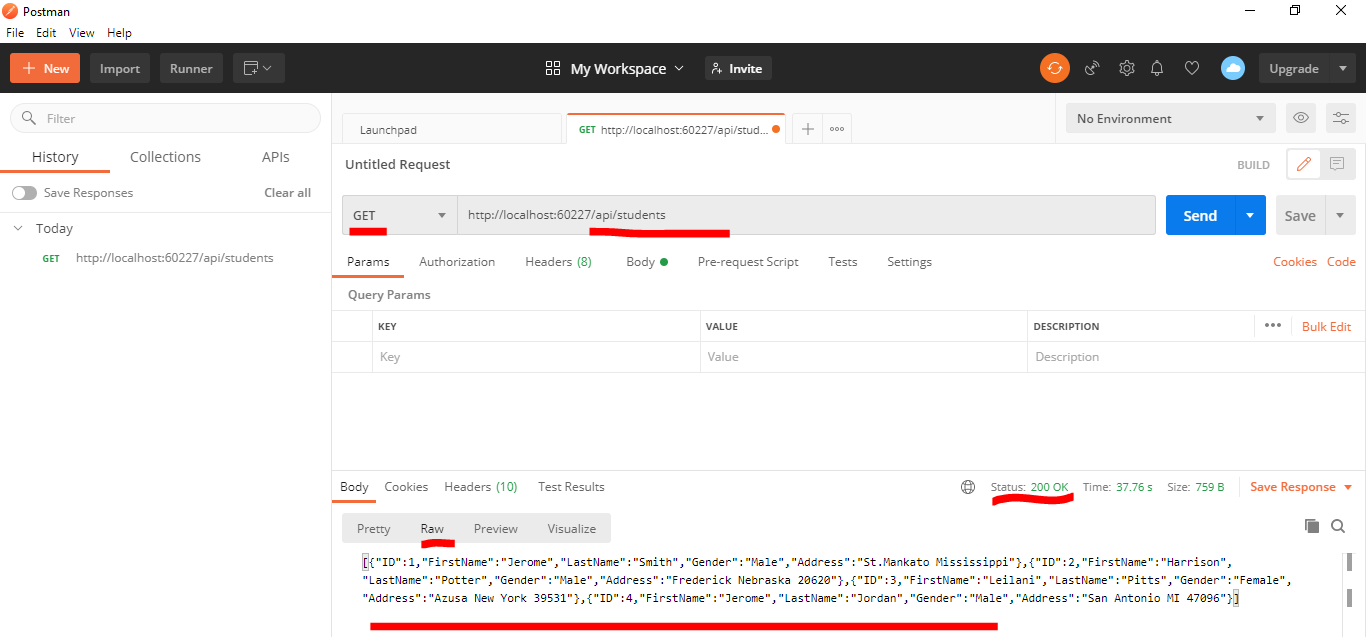
##### Task 5 - ****Implementing the GET method in ASP.NET Web API****

##### Add the following code in StudentsController.cs

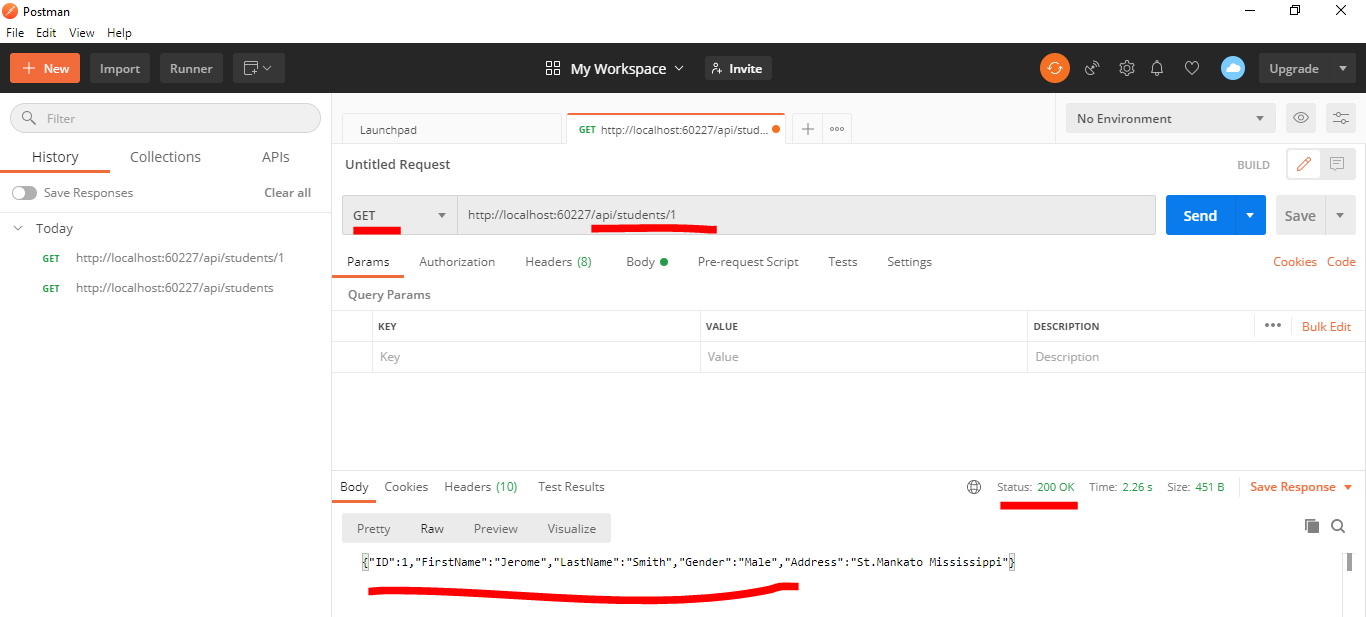
##### 

**Task 6 – Run your application and observe the result using a Web API client tool like Postman**

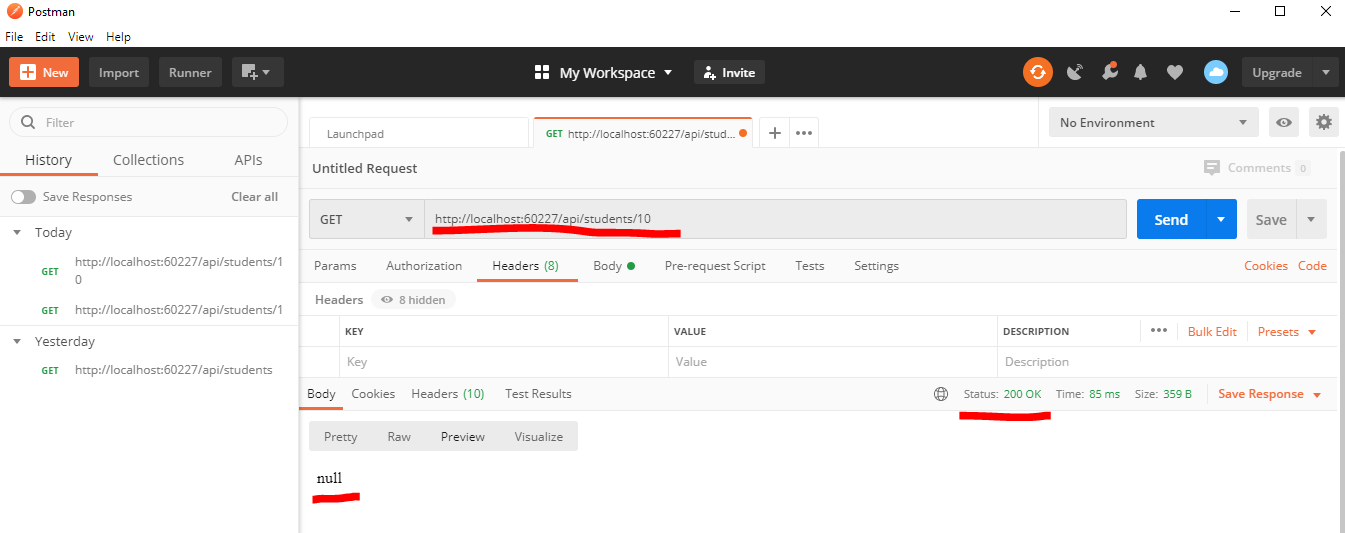
1. Select the **Get** requestand type the **URI** and click **Send.**



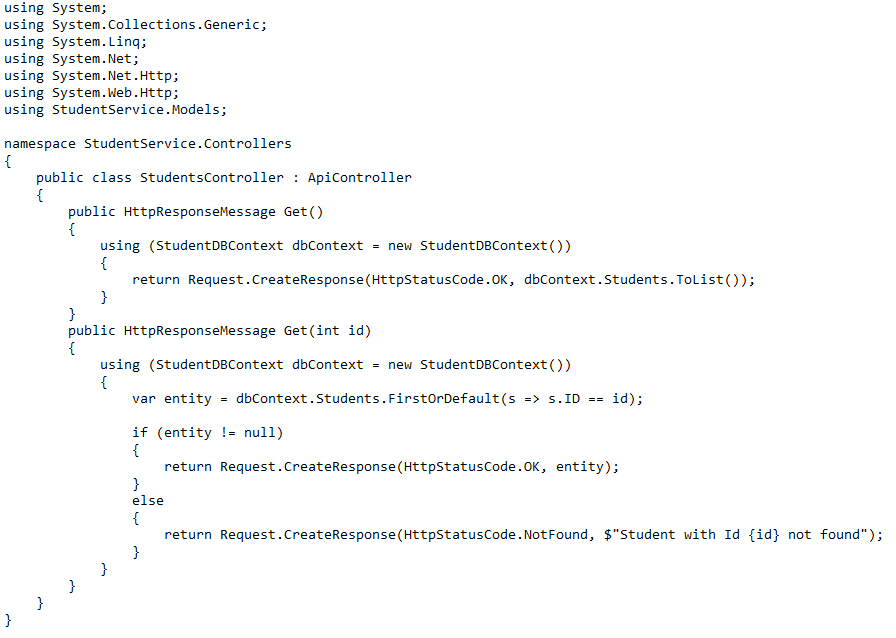
1. Verify the response data.
2. Make changes to the URI and verify whether you get the expected response. Here api/students/1 indicate the student record having student id 1.



1. You will get null value for invalid id as shown in the below image.



1. Now modify your controller logic so that you will get a user-friendly message as response.



1. After updating the controller logic, rebuild and then restart your application.
2. Send a GET request with an invalid id using Postman and inspect the response from the server.

