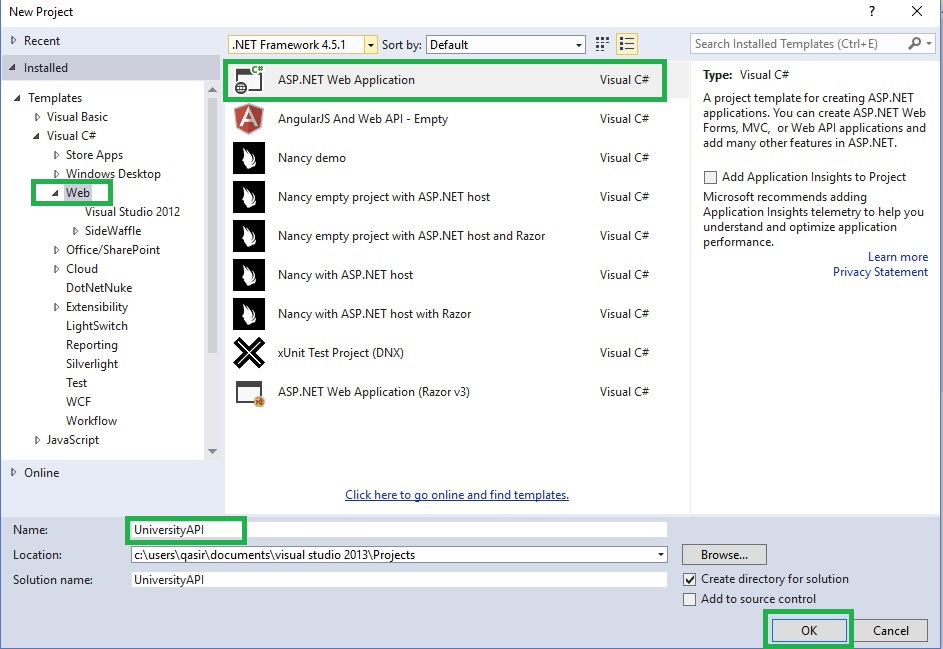
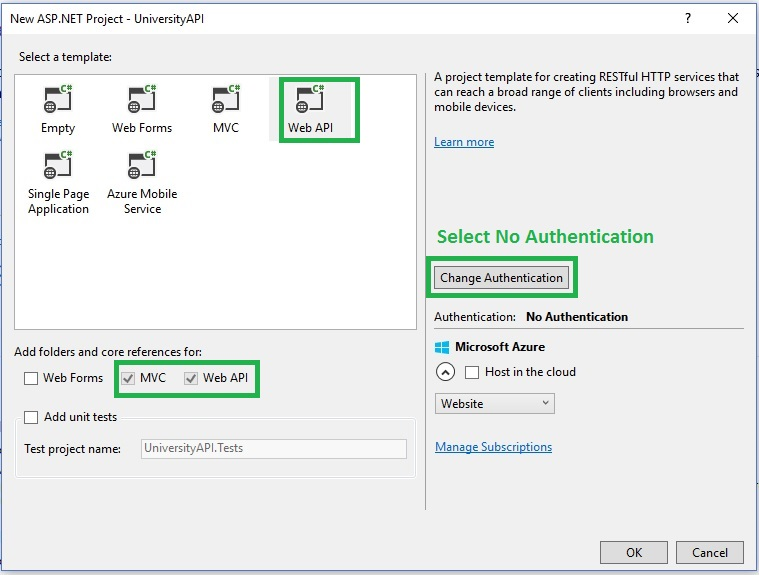
**University-Web API with EF 6.1.3**

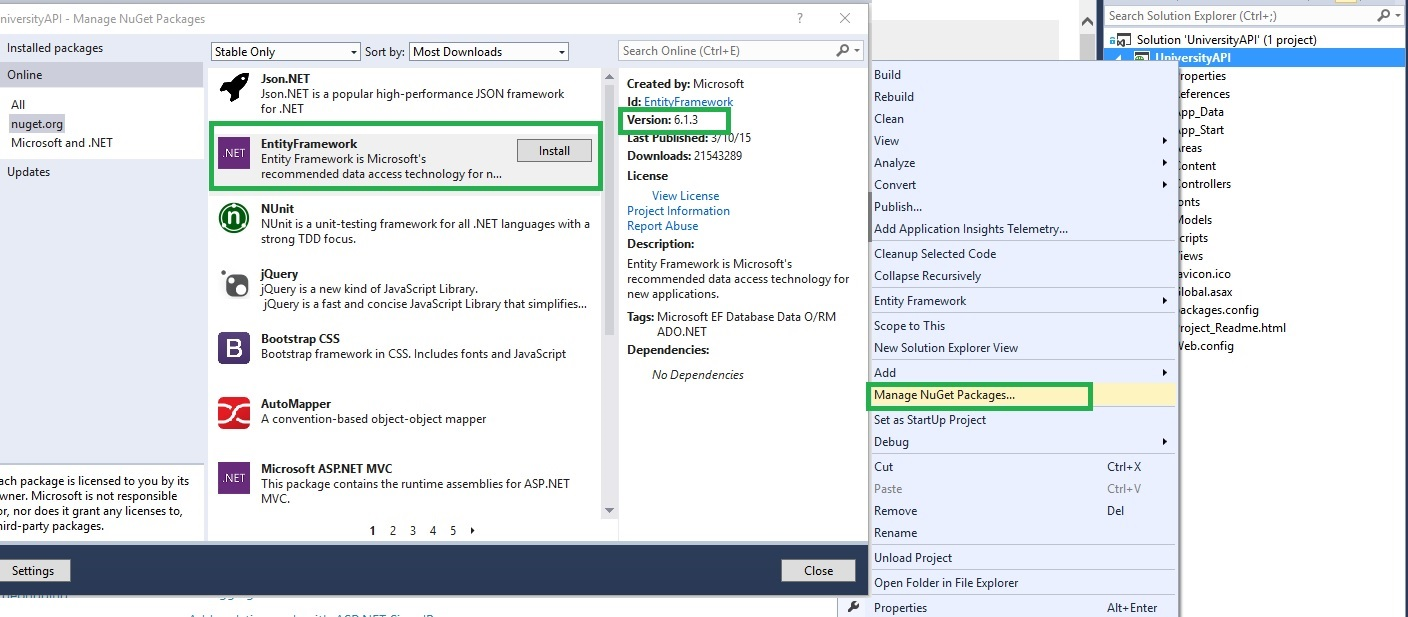
You can build Web API in VS2013 and EF 6.1.3 code first from existing database  
  
**Easy steps to setup Web API with EF code first in VS 2013**

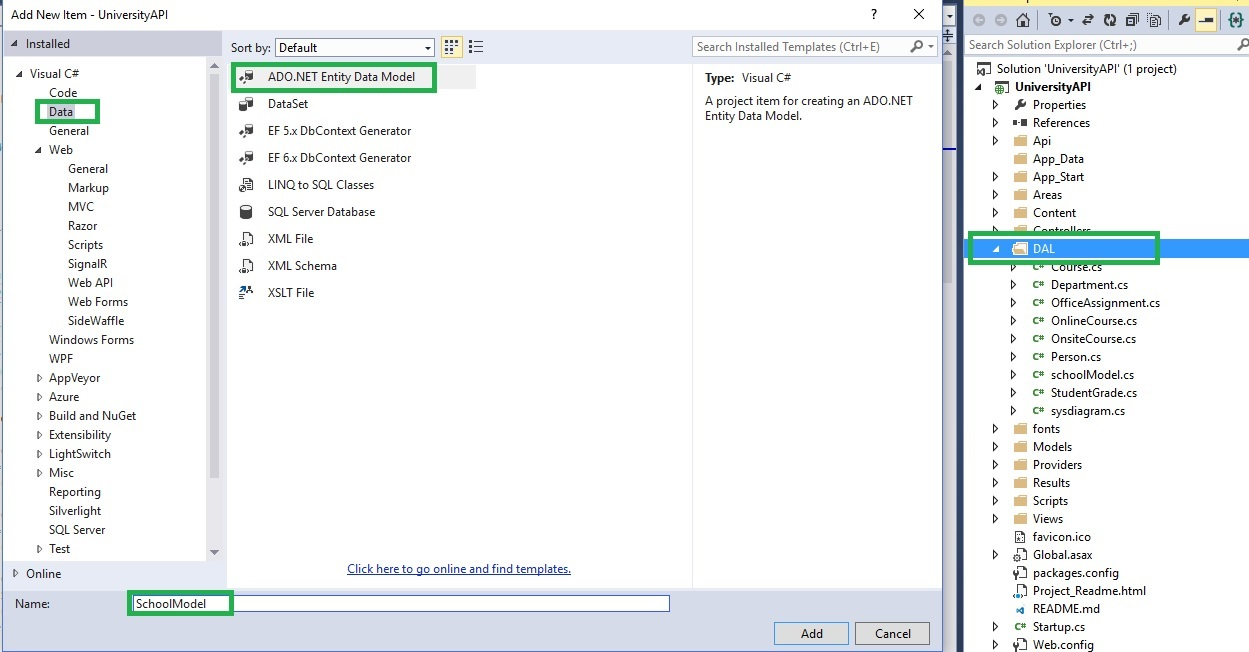
Restore data base in SQL Server 2014 (Files provided in Database folder)

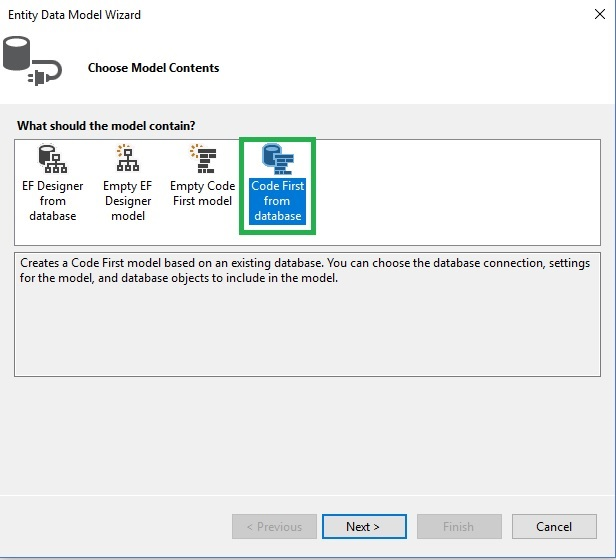
**Steps for New Project in VS 2013**

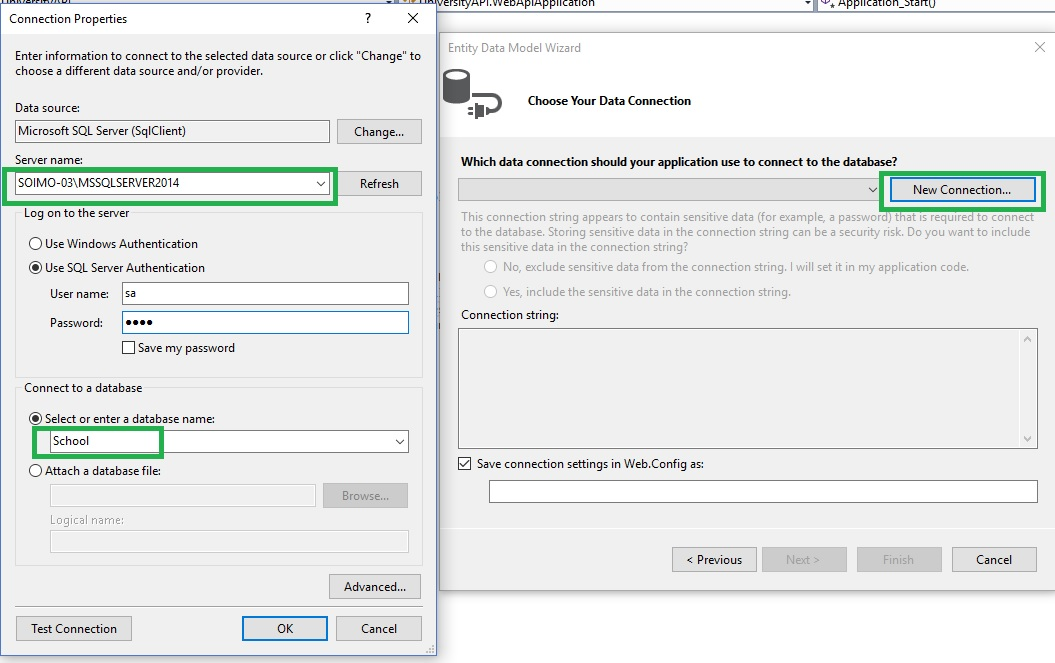
1 - Create New Web project  


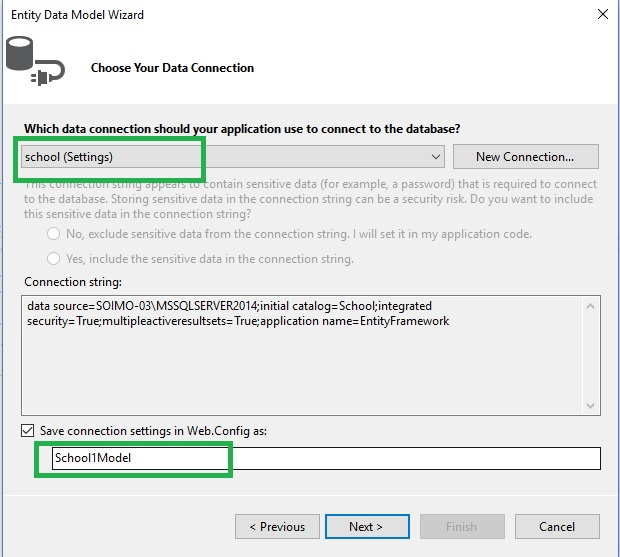
2 - Select Web API with MVC and no authentication  


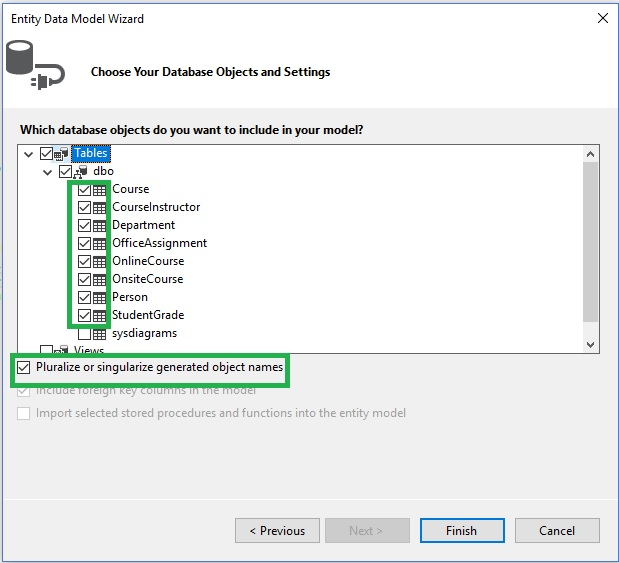
3- Install EF 6.1.3 for Code first from existing database  


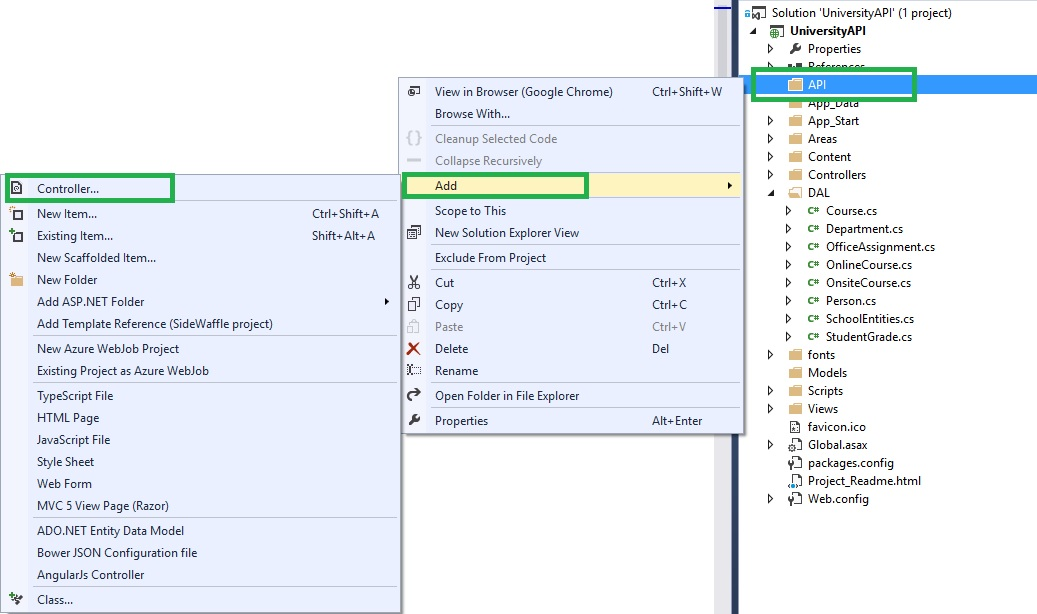
4 - Create a new folder DAL, and right click select Add New Item  
Select Data option with ADO.NET Entity Data Model  


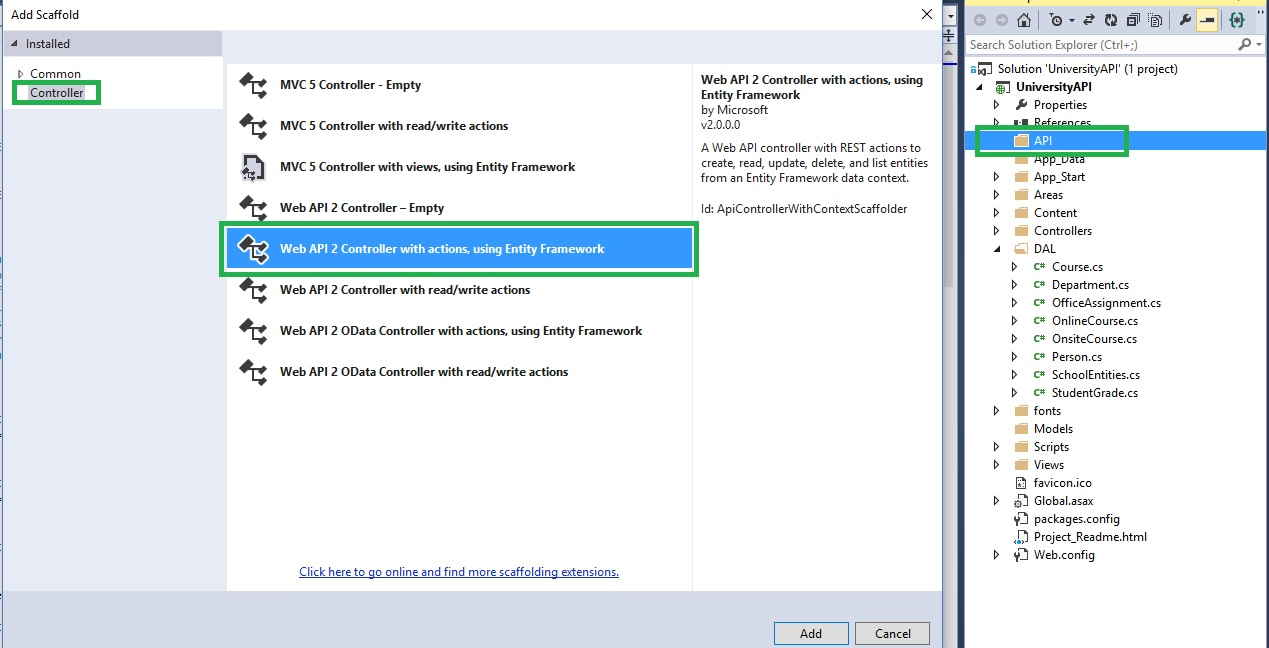
5 - Select Code first from Database  


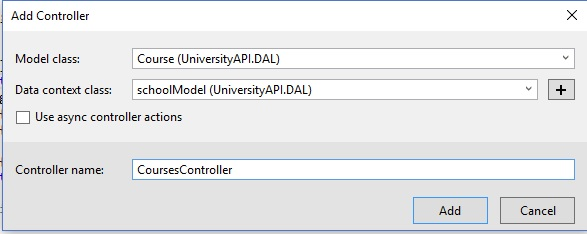
6- Create new connection with SQL server, and select School database  




7- Select tables which you want to include in model  


8- Create an API folder where you will put API controllers  
Right click on API folder to add Controller  


9- In controller, select Web API 2 Controller with actions, using Entity Framework  


10 - For Course table create controller.  


11 - **If receive an error then try to rebuild project before adding controller**

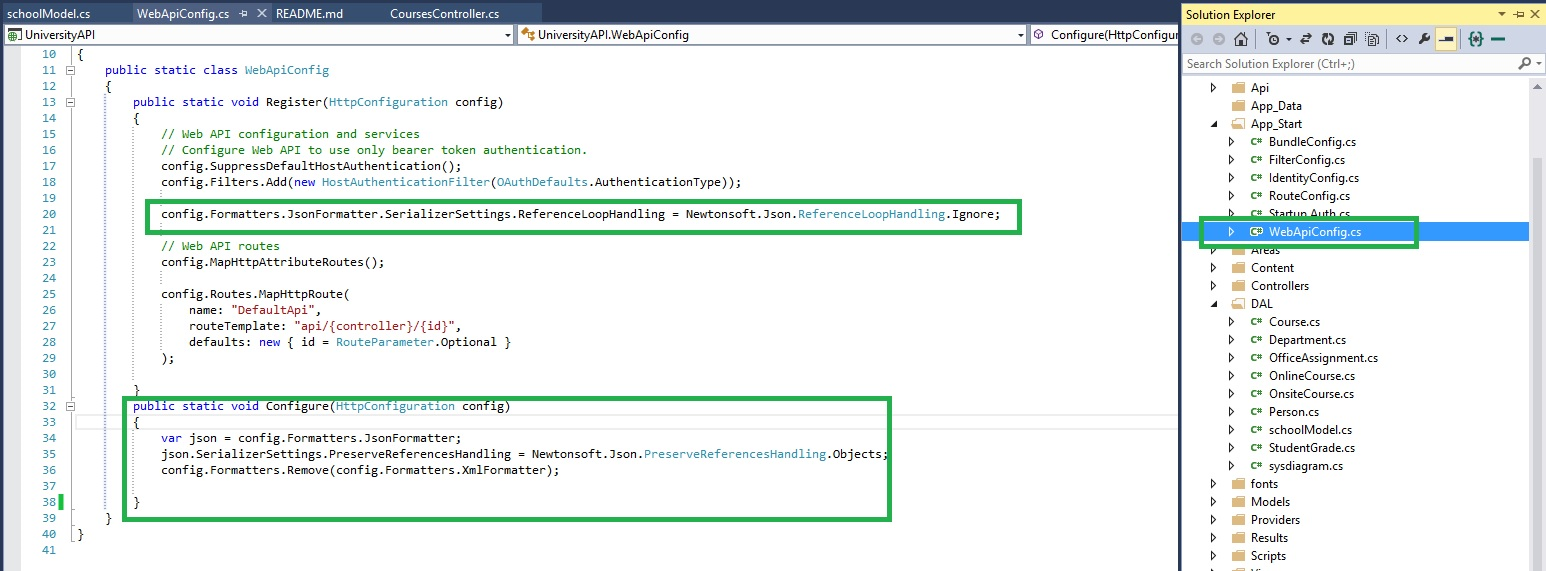
12- Now Web API project is ready to test  
- Before run, for re-reference looping add following line at App\_Start/WebApiConfig/Register  
  
**config.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;**  
  
- For JSON results add following Configure function at App\_Start/WebApiConfig/Register  
**public static void Configure(HttpConfiguration config)**

**{**

**var json = config.Formatters.JsonFormatter;**

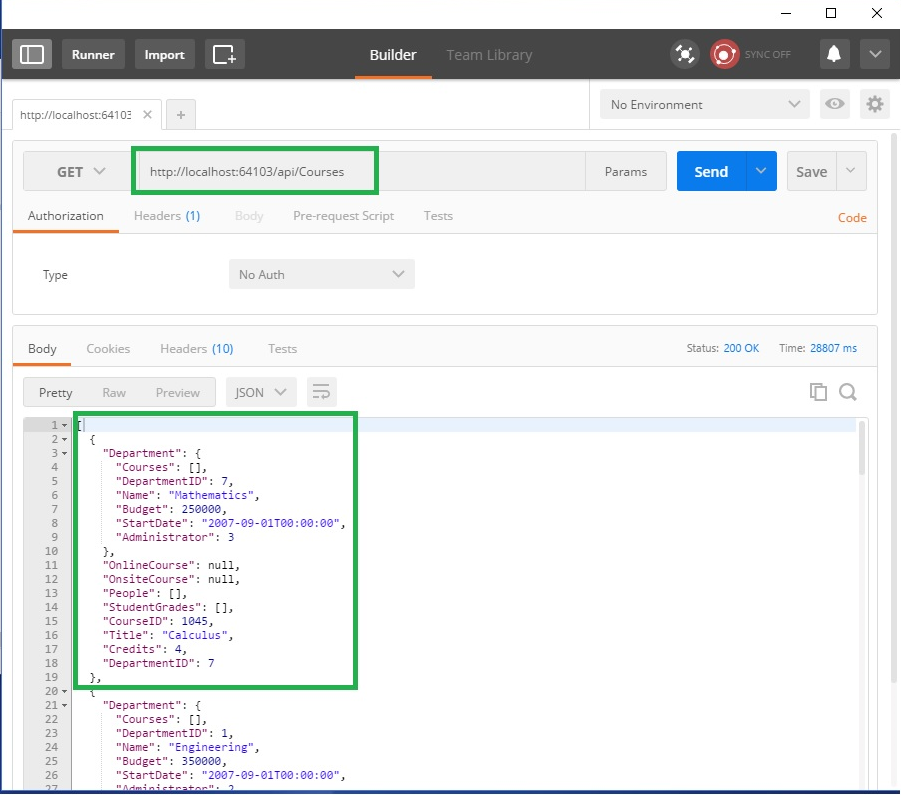
**json.SerializerSettings.PreserveReferencesHandling = Newtonsoft.Json.PreserveReferencesHandling.Objects;**

**config.Formatters.Remove(config.Formatters.XmlFormatter);**

**}**  
  


13- For Lazy Loading use following code at DAL/SchoolModel.cs/constructor  
  
**this.Configuration.LazyLoadingEnabled = false;**

**14 -** Run project and you will be able to see running Web API like **http://localhost:64238/**   
In chrome extension "Postman" you can see JSON results by browsing URL : **http://localhost:64238/api/Courses**



15- If you like to add department with this JSON results, the use following code in Course API controller

**return db.Courses.Include(d => d.Department);**