

1. Create ASP.Net MVC application
2. Right Click the controller folder and from ‘Add Scaffold’ section select the scaffold template “MVC 5 Controller with read/write actions”
3. Name the Controller as **ProductController** and click on ‘Add’
4. Now a ProductController will be created with Action method for insert, update, delete and view etc.
5. First let’s take Index action and we will understand how to list or view all the products in our table
6. All the action methods are **HttpGet** request by default, hence we will put HttpGet attribute before our action method

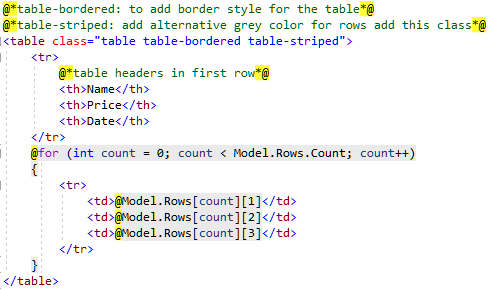


1. Create a connection string and then declare the required namespaces

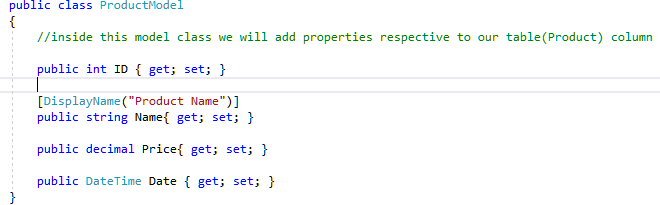
System.Data and

System.Data.SqlClient

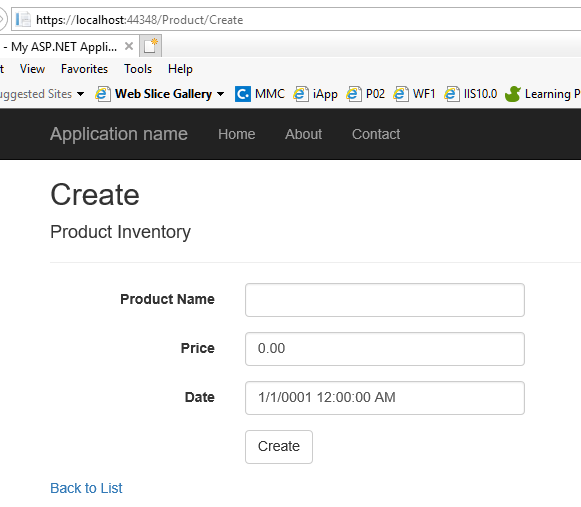
1. In Index action method, populate the DataTable from the database. Pass the DataTable as parameter to the View
2. Right click inside action method and select ‘Add View’, keep the name as it is as **Index**
3. Index.cshtml page, declare the model data type as System.Data.DataTable since we are passing DataTable from Index action method
4. Create html <table> and decorate with bootstrap methods



1. To run the application Hold Ctrl + F5
2. Let’s get rid of the Details action method as we don’t need this action method
3. Let’s focus on the Create action method which is of type **HttpGet**. Using this action method, we will create a form to enter product details
4. Before that right click on Models folder and add a model for our products
5. Right click model and add class ‘ProductModel’ inside this class we will add properties respective to our Product table columns



1. Now let’s go back to product controller, now pass object of the class ProductModel (also import namespace for the model)
2. To add a view for this action method right click inside the action method ‘Add Vew’
3. Keep the name of the view as it is ‘Create’ and from Template select the name as ‘Create’ and them from model class select ProductModel (LearnMVC.Models)
4. Using the scaffolding mechanism MVC creates a new View using Template and Model class
5. Run the application name from Create.cshtml, hold Ctrl + F5



1. We need to make few modifications now, like in Index view we need to add a new hyperlink to add a new Product (redirect the user to Create Product Inventory page)
2. In ProductModel.cs class add the decorator [DisplayName(“Product Name”)] to the entity ProductName (also add link to add the required namespace: System.Component model)
3. Navigate to Index.cshtml page and add an anchor tab



here the first parameter will be action name: Create then second parameter as controller name: Product

1. When we post a form, it will look for action method of type **HttpPost**

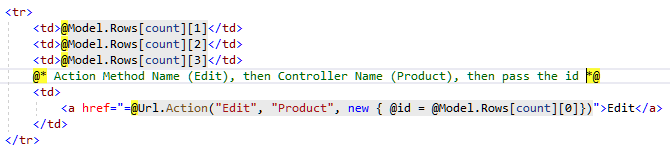
In our form we have two action methods Create with HttpPost and HttpGet



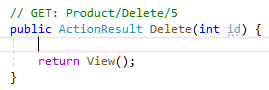


We need to refer the action method Create with HttpPost attribute

1. First change the action method parameter with product model (data sent from the form will be binded to this productModel)
2. Write code to post the data from controller to database using model
3. After data add, redirect to index action method (navigate to Index page)
4. Build the solution and run the application from ProductController (Ctrl + F5)
5. In order to perform Edit or Update operation we have two action methods **Edit** one is of type HttpGet and other is of type HttpPost
6. In case of HttpGet action method, we have the parameter id of the type integer. When the url /Product/Edit/5
7. 5 will be taken as id and the details of the product with id 5 will be taken. Write in Edit (int id) action method
8. Using SqlDataAdaptor get the data from database for specific product id and assign it to Product Model
9. Till now we have not added a View for Edit action method, lets add a View
10. View name **Edit**, Template **Edit** and Model Class **ProductModel**
11. Save the project and build the solution - <https://localhost:44348/Product/Edit/2007>
12. Navigate to the Index view and add an additional column for ‘Edit’ link button
13. @Url.Action (Action Method Name, Controller Name, Parameter Name (product id))



1. Now load the index view (Ctrl + F5)
2. In Edit view (<https://localhost:44348/Product/Edit/2007>) if Save button is clicked it will post the form to HttpPost action method
3. In HttpPost action result ‘Edit’, pass the Product model and save the information to database and then redirect to Index action method
4. Finally, we will plan to delete a record from the database



1. Add to code to delete the record from database and then redirect the user to Index action method
2. In Index action method, add a link to delete the record
3. Save Project and build solution