Step-by-Step Guide to perform CRUD operations using Entity Framework Core with SQL Server in an ASP.NET Core MVC Application

---

1. **Install Required Packages**

- `Microsoft.EntityFrameworkCore` → Base package.
- `Microsoft.EntityFrameworkCore.SqlServer` → SQL Server provider.
- `Microsoft.EntityFrameworkCore.Tools` → Migration/scaffolding commands.

2.1. **Create the Model**

```
public class Product
{
    public int Id { get; set; }   // Primary Key
    public string Name { get; set; }
    public decimal Price { get; set; }
    public string Category { get; set; }
}
```

2.2. **Set up the DbContext**

```
using Microsoft.EntityFrameworkCore;

public class ApplicationDbContext : DbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }

    public DbSet<Product> Products { get; set; }
}
```

---

3. **Configure Connection String**

Open `appsettings.json` and add a connection string:

```
"ConnectionStrings": {
  "DefaultConnection": "Server=DESKTOP-
B3L5S5L\\SQLEXPRESS;Database=ProductsDb;Integrated
Security=True;Trusted_Connection=True;TrustServerCertificate=True"
}
```

4. **Register DbContext in Program.cs**

```
builder.Services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"
)));
```

5. **Run EF Core Migrations**

Run the following commands in **Package Manager Console** (or CLI):

```
Add-Migration InitialCreate
Update-Database
```

This will create the `Products` table in the database.

6. **Create ProductsController and Inject the ApplicationDbContext**

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

public class ProductsController : Controller
{
    private readonly ApplicationDbContext _context;
    public ProductsController(ApplicationDbContext context)
    {
        _context = context;
    }
}
```