CO550 – Web Applications

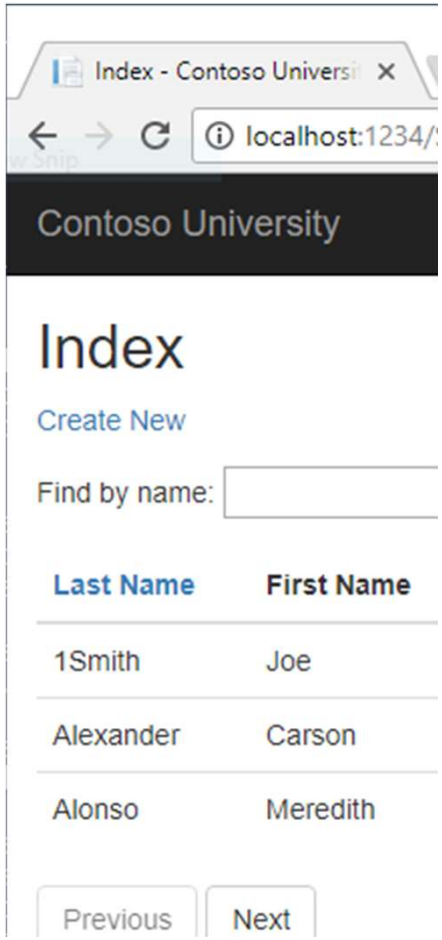
UNIT 8 – LINQ, Database migrations

Tutorial Recap

Step 1 of our ASP.NET Core Razor Pages Tutorial covered…

https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/intro?view=aspnetcore-2.1&tabs=visual-studio

- Manipulating template files
- Setting up a data model
- Scaffolding the data model
- Create the database with `EnsureCreated`
- Initialising the database with seed data



Step 2

https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/crud?view=aspnetcore-2.1

- Understanding the CRUD pages created
- Asynchronous calls
- Adding related data to a Student Details page (get enrolled courses of a student)
- Overposting and basic security

Tutorial: Step 3

Sorting, Searching, Pagination

bucks new university

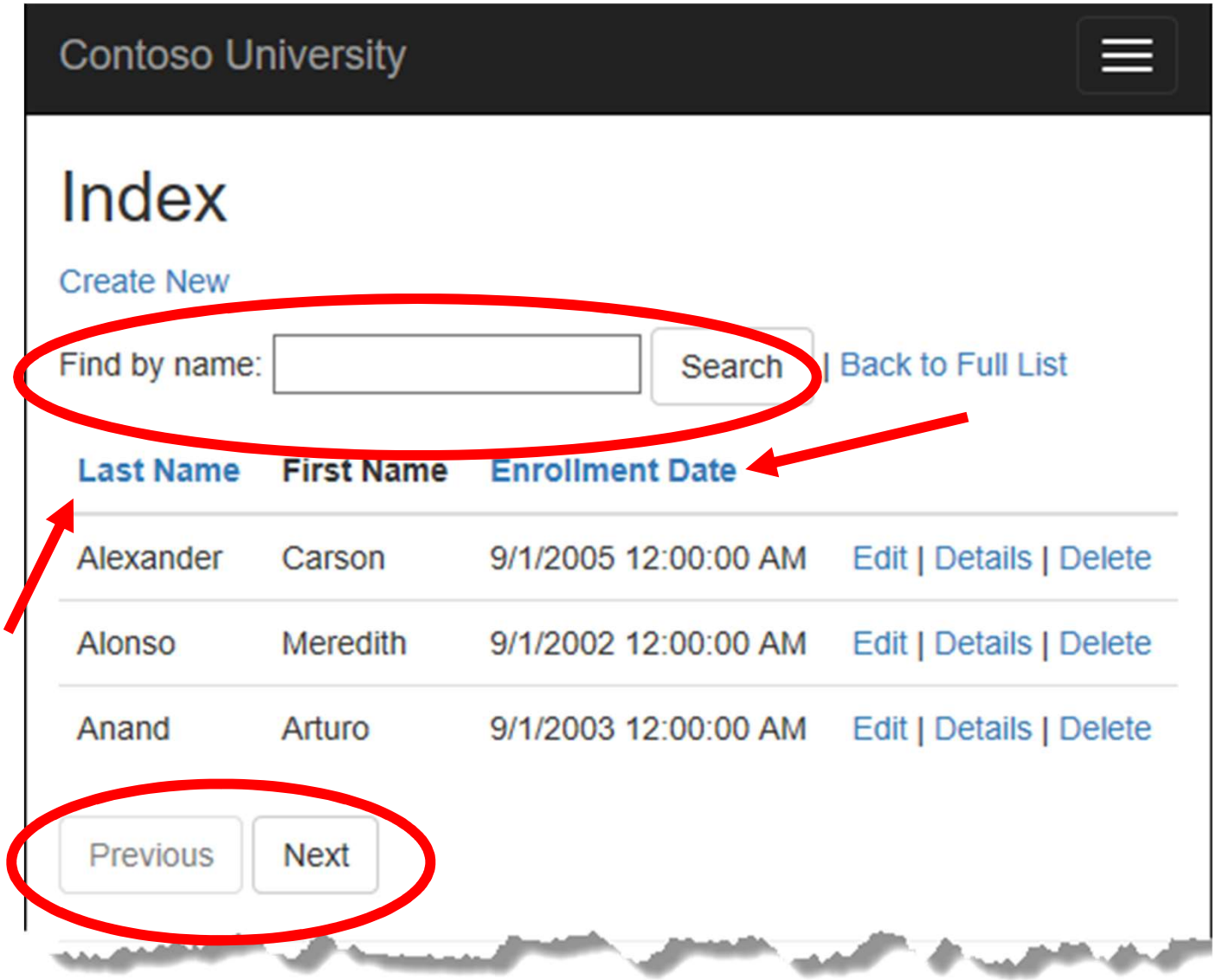
Now For… Step 3

https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/sort-filter-page?view=aspnetcore-2.1

- Sorting, Searching, Filtering the Student Data
- LINQ
- Pagination using ASP.NET and LINQ
- An alternative approach to pagination using jQuery Data Tables (not actually covered in the Microsoft tutorial)
- A student statistics page (again, using LINQ)

bucks new university

What we're aiming for…



bucks new university

We need to update the page model

C#

```
public class IndexModel : PageModel
{
    private readonly SchoolContext _context;

    public IndexModel(SchoolContext context)
    {
        _context = context;
    }

    public string NameSort { get; set; }
    public string DateSort { get; set; }
    public string CurrentFilter { get; set; }
    public string CurrentSort { get; set; }
```

Students/Index.cshtml.cs

The base LINQ query

Applying sort order to the LINQ query

Default sort order

Go get the students as a list

C#

```
public async Task OnGetAsync(string sortOrder)
{
    NameSort = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
    DateSort = sortOrder == "Date" ? "date_desc" : "Date";

    IQueryable<Student> studentIQ = from s in _context.Student
                                    select s;

    switch (sortOrder)
    {
        case "name_desc":
            studentIQ = studentIQ.OrderByDescending(s => s.LastName);
            break;
        case "Date":
            studentIQ = studentIQ.OrderBy(s => s.EnrollmentDate);
            break;
        case "date_desc":
            studentIQ = studentIQ.OrderByDescending(s => s.EnrollmentDate);
            break;
        default:
            studentIQ = studentIQ.OrderBy(s => s.LastName);
            break;
    }

    Student = await studentIQ.AsNoTracking().ToListAsync();
}
```

Students/Index.cshtml

- We then need to update the page HTML…

```
<table class="table">
    <thead>
        <tr>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.NameSort">
                    @Html.DisplayNameFor(model => model.Student[0].LastName)
                </a>
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Student[0].FirstMidName)
            </th>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.DateSort">
                    @Html.DisplayNameFor(model => model.Student[0].EnrollmentDate)
                </a>
            </th>
            <th></th>
        </tr>
```

- Note: we are using C# (Razor syntax) to execute server side code in our “page” or what would be the “view” in an MVC application

bucks new university

So what is LINQ?

- LINQ = “Language Integrated Query”
- *We use the term **language-integrated query** to indicate that query is an integrated feature of the developer's primary programming languages (e.g. C#)*
- This means we can write queries in C# and never have to worry about the underlying storage mechanism for the data (e.g. Database, XML, other..)

Resources:

- https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/ (gives a good overview)
- https://msdn.microsoft.com/en-gb/library/bb308959.aspx (more detail)
- http://www.tutorialsteacher.com/linq/what-is-linq

bucks new university

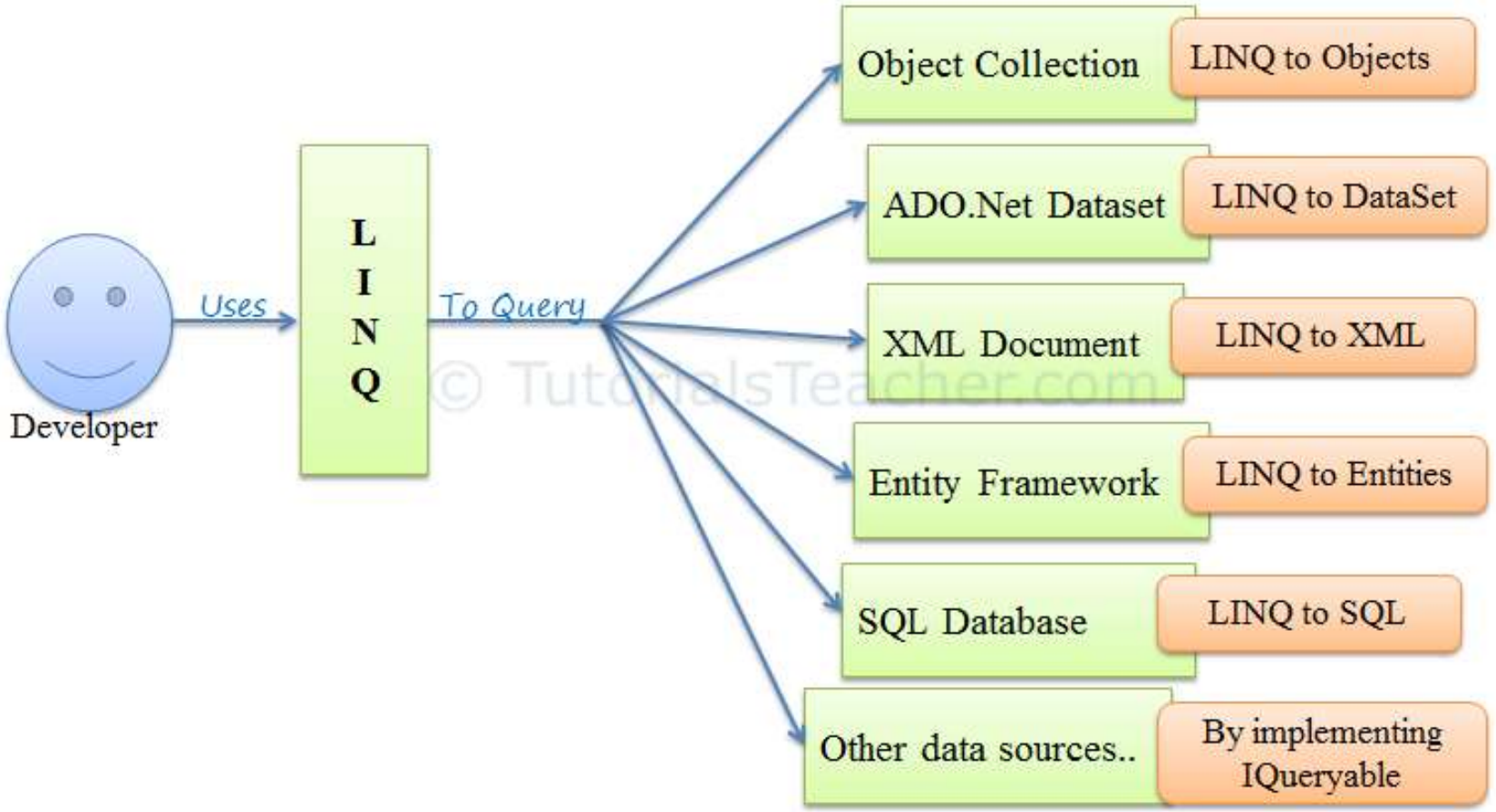
So what is LINQ?

- LINQ is uniform query syntax in C# … used to save and retrieve data from different sources.
- It is integrated in C#, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources.

Source: http://www.tutorialsteacher.com/linq/what-is-linq

So what is LINQ?

Source: http://www.tutorialsteacher.com/linq/what-is-linq



bucks new university

A LINQ Example

C#

```
class LINQQueryExpressions
{
    static void Main()
    {

        // Specify the data source.
        int[] scores = new int[] { 97, 92, 81, 60 };

        // Define the query expression.
        IEnumerable<int> scoreQuery =
            from score in scores
            where score > 80
            select score;

        // Execute the query.
        foreach (int i in scoreQuery)
        {
            Console.Write(i + " ");
        }
    }
}
// Output: 97 92 81
```

A note on IQuerables

- When an IQueryable is created or modified, no query is sent to the database.
- The query isn't executed until the IQueryable object is converted into a **collection**.
- IQueryables are converted to a collection by calling a method such as **ToListAsync**.
- Much better for performance!

```
C#

Student = await studentIQ.AsNoTracking().ToListAsync();
```

IEnumerable Vs IQueryable

Many developers get confused between IEnumerable and IQueryable. When it comes to writing code, both look very similar.

However, there are many differences between them which need to be taken care of while writing code.

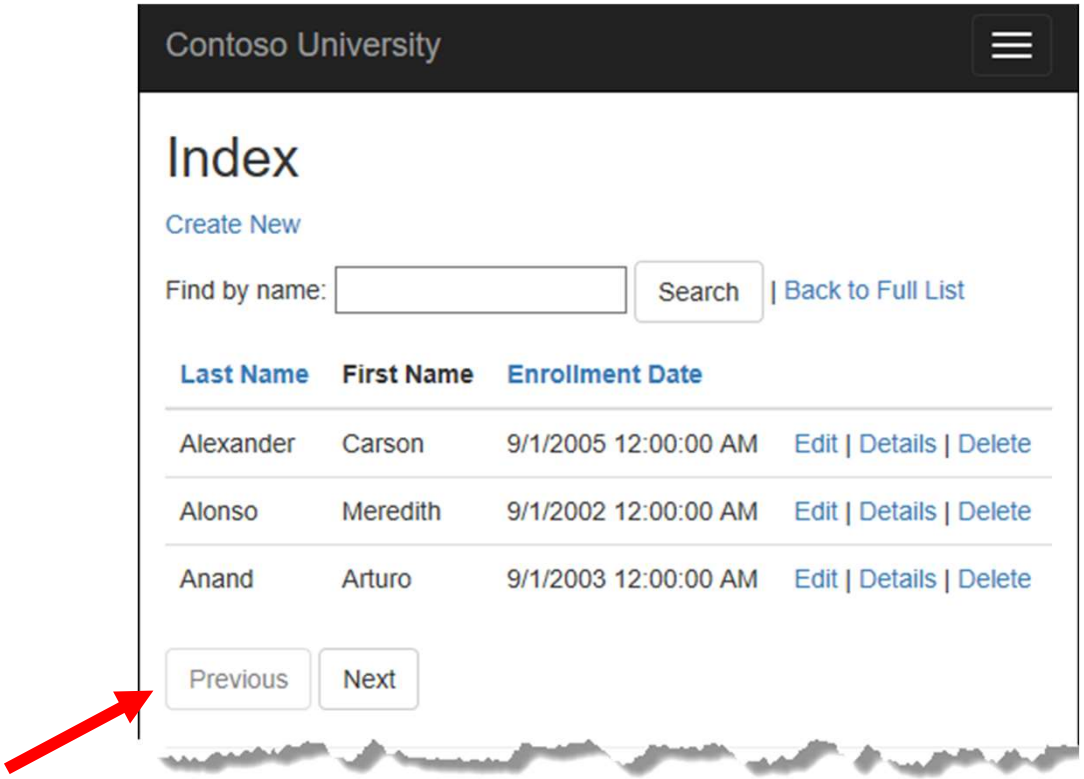
Source: https://www.codeproject.com/Articles/732425/IEnumerable-Vs-IQueryable

bucks new university

	IEnumerable	IQueryable
Namespace	`System.Collections` Namespace	`System.Linq` Namespace
Derives from	No base interface	Derives from `IEnumerable`
Deferred Execution	Supported	Supported
Lazy Loading	Not Supported	Supported
How does it work	While querying data from database, `IEnumerable` executes `select` query on server side, load data in-memory on client side and then filter data. Hence does more work and becomes slow.	While querying data from database, `IQueryable` executes `select` query on server side with all filters. Hence does less work and becomes fast.
Suitable for	LINQ to Object and LINQ to XML queries	LINQ to SQL queries
Custom Query	Doesn't support	Supports using `CreateQuery` and `Execute` methods
Extension method parameter	Extension methods supported in `IEnumerable` takes functional objects.	Extension methods supported in `IEnumerable` takes expression objects, i.e., expression tree.
When to use	When querying data from in-memory collections like `List`, `Array`, etc.	When querying data from out-memory (like remote database, service) collections.
Best Uses	In-memory traversal	Paging

Pagination

- We next add paging links to the Students index page
- The paging will need to work properly even if we sort the data on different columns or carry out a search query.



bucks new university

A Lazy Approach to Pagination

- Using jQuery Data Tables
- Include the client-side libraries in the main view/page template

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<link href="~/lib/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />

<link href="https://cdn.datatables.net/1.10.15/css/dataTables.bootstrap.min.css" rel="stylesheet" />
<link href="https://cdn.datatables.net/responsive/2.1.1/css/responsive.bootstrap.min.css" rel="stylesheet" />

<script src="https://cdn.datatables.net/1.10.15/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.10.15/js/dataTables.bootstrap4.min.js "></script>
```

bucks new university

jQuery Data Tables

```
<div class="container">
   <br />
   <div style="width:90%; margin:0 auto;">
      <table id="example" class="table table-striped table-bordered dt-
responsive nowrap" width="100%" cellspacing="0">
         <thead>
            <tr>
               <th>CustomerID</th>
               <th>Name</th>
               <th>Address</th>
               <th>Country</th>
            </tr>
         </thead>
         <tbody>
           // foreach loop here to output row data
         </tbody>
      </table>
   </div>
</div>
```

jQuery Data Tables

Initialise the data table …

```
<script>
    $(document).ready(function ()
    {
        $('#example').dataTable( {
        });
    });
</script>
```

bucks new university

Working examples

https://datatables.net/examples/basic_init/zero_configuration.html

https://codepen.io/spenser/pen/wKdzay

Question: is there anything wrong with loading ALL of the data on page load?

bucks new university

Tutorial: Step 4

Database Migrations

bucks new university

Now For... Step 4

https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/migrations?view=aspnetcore-2.1&tabs=visual-studio

- What are Migrations?
- Why are they useful?
- Removing **EnsureCreated** and using Migrations instead
- Using the **Package Manager Console** (PMC)

bucks new university

Database Migrations

- Our current approach to managing database changes is:
  - The DB is dropped.
  - EF creates a new one that matches the model.
  - The app seeds the DB with test data.

- Problem with this?

- Rather than dropping and recreating the DB when the data model changes, **migrations** update the schema and retain existing data.

Database Migrations

How does it work?

- First we'll delete our database using either…
  - the Package Manager Console (PMC)
  - SQL Server Object Explorer
- Add first migration (with a name)
- Update database
- A new class will be created with up() and down() methods

Migrations<timestamp>_InitialCreate.cs

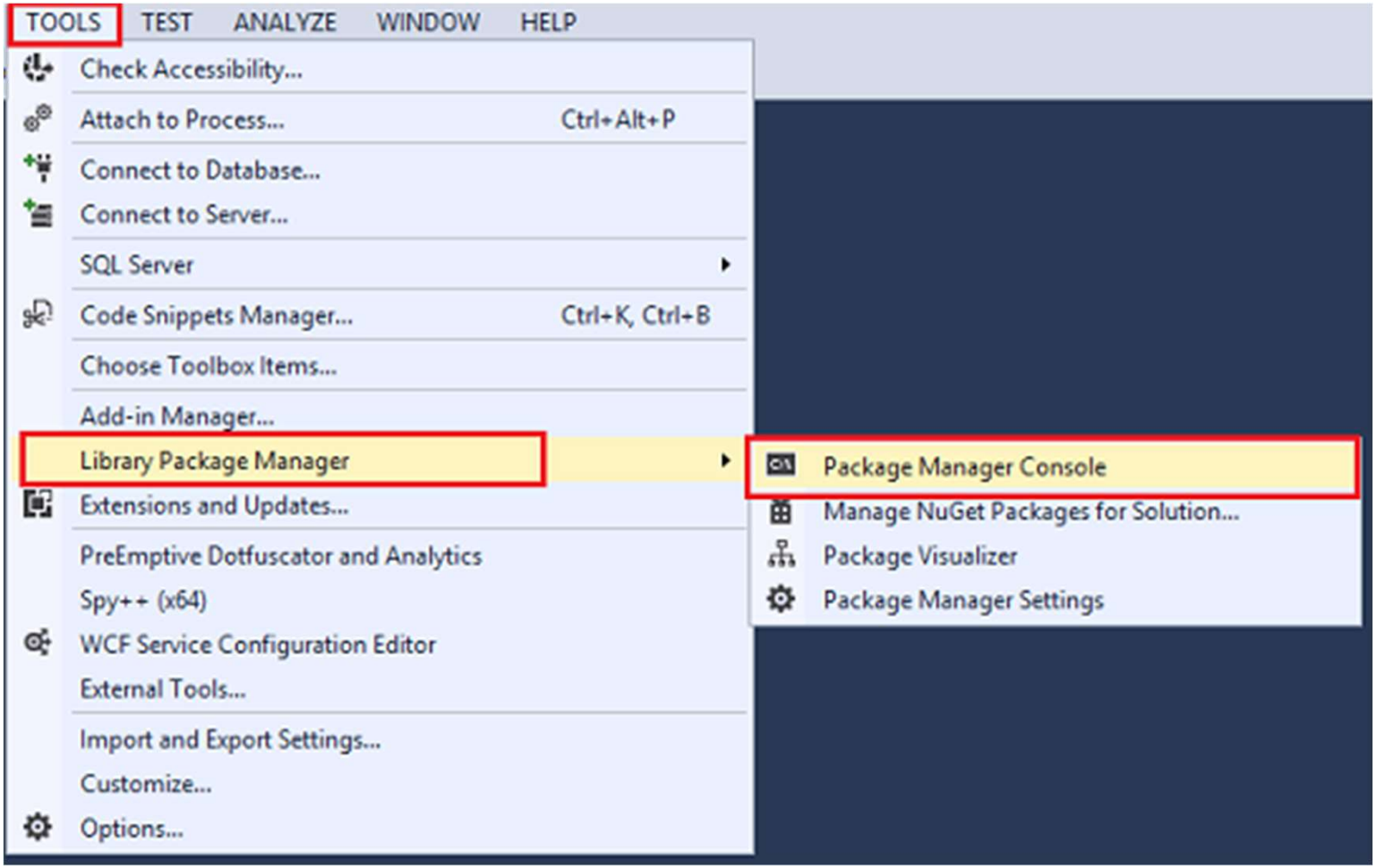
PMC

```
Drop-Database
```

PMC

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

Package Manager Console (PMC)



bucks new university

Database Migrations

- Migrations create a *snapshot* of the current database schema in *Migrations/SchoolContextModelSnapshot.cs*.

- When you add a migration, Entity Framework (EF) determines what changed by comparing the data model to the snapshot file.

https://www.youtube.com/watch?v=FbX2NX9Xdr8

bucks new university

Further Reading Reminders

bucks new university

ASP.NET Core Video Tutorials

https://www.youtube.com/playlist?list=PLDmvslp_VR0x2CmC6c4AZhZfYX7G2nBIo

1 

Introduction to ASP.NET Core 2 | Environment Setup | Part 1 | Eduonix
Eduonix Learning Solutions

2 

Introduction to ASP.NET Core 2 | Overview | Part 2 | Eduonix
Eduonix Learning Solutions

3 

Introduction to ASP.NET Core 2 | Application Structure | Part 3 | Eduonix
Eduonix Learning Solutions

4 

Introduction to ASP.NET Core 2 | Life Cycle Of An App | Part 4 | Eduonix
Eduonix Learning Solutions

5 

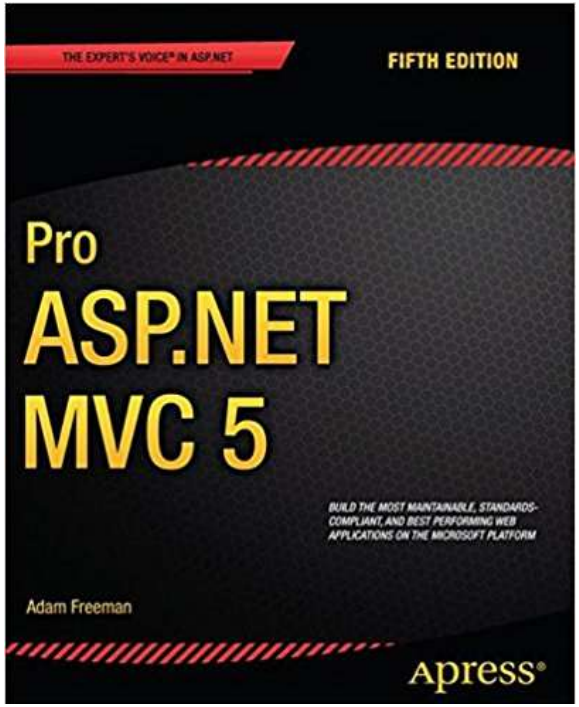
ASP.NET Core 2 Tutorial | Middleware | Part 5 | Eduonix
Eduonix Learning Solutions

bucks new university

Further Reading

Pro ASP.NET MVC 5 (Fifth Edition)

Available online (and in print) via BNU Library



Chapter 1: Putting ASP.NET MVC in Context (p. 1-10)

Chapter 3: The MVC Pattern (p. 51-66)

- The history of MVC
- Understanding the MVC pattern
- Loose coupling
- Automated testing

bucks new university