Manish Jaiswal

LINQ Series

MASTERING

Advanced LINQ Techniques

swipe

Manish Jaiswal

LINQ Series

01

LINQ with Reflection

Reflection allows you to inspect and interact with object types at runtime. Combining it with LINQ offers dynamic querying capabilities.

swipe

Manish Jaiswal

LINQ Series

02

Example

This example retrieves and prints all string properties of the Person class using reflection and LINQ.

```
using System;
using System.Linq;
using System.Reflection;

var properties = typeof(Person).GetProperties()
    .Where(p => p.PropertyType == typeof(string))
    .Select(p => p.Name);

foreach (var prop in properties)
{
    Console.WriteLine(prop);
}
```

swipe

Manish Jaiswal

LINQ Series

03

Expression Trees & Dynamic LINQ

Expression trees represent code in a tree-like data structure, making it possible to construct and manipulate queries dynamically.

swipe

Manish Jaiswal

LINQ Series

04

Example

This example creates a simple expression tree to evaluate if a number is greater than 5.

```
1  using System;
2  using System.Linq.Expressions;
3
4  Expression<Func<int, bool>> expr = num => num > 5;
5  Func<int, bool> func = expr.Compile();
6
7  bool result = func(10); // True
8  Console.WriteLine(result);
```

swipe

Manish Jaiswal

LINQ Series

05

Dynamic LINQ Query

Dynamic LINQ enables building queries dynamically based on user input or other runtime conditions.

swipe

Manish Jaiswal

LINQ Series

06

Example

This example demonstrates a dynamic query that filters Persons based on age and first name.

```
using System.Linq.Dynamic.Core;

var query = context.Persons
    .Where("Age > @0 and FirstName == @1", 30, "John")
    .Select("new (FirstName, LastName)");

foreach (var person in query)
{
    Console.WriteLine($"{person.FirstName} {person.LastName}");
}
```

swipe

Manish Jaiswal

LINQ Series

07

LINQ with .NET Features

LINQ can be combined with other .NET features such as async programming, parallel processing, and more to enhance performance and functionality.

swipe

Manish Jaiswal

LINQ Series

08

Example

This example uses LINQ with asynchronous programming and parallel processing to filter even numbers.

```
using System;
using System.Linq;
using System.Threading.Tasks;

var numbers = Enumerable.Range(1, 100);

var evenNumbers = await Task.Run(() =>
    numbers.AsParallel()
           .Where(n => n % 2 == 0)
           .ToList());

foreach (var number in evenNumbers)
{
    Console.WriteLine(number);
}
```

swipe