

Making Expression Trees Work for Us



Chris B. Behrens

@chrisbbehrens

Where We're Headed

A sample web interface

Using the data of the
passenger manifest
of the Titanic



Demo



Look at the base sample data I got off the Internet

Look at how I've wrapped this in some simple code

View the basic application to work with it

Talk about why I implemented it the way I did



Why Not Entity Framework?

In a real application, I would use EF

IEnumerable and IQueryable are equivalent

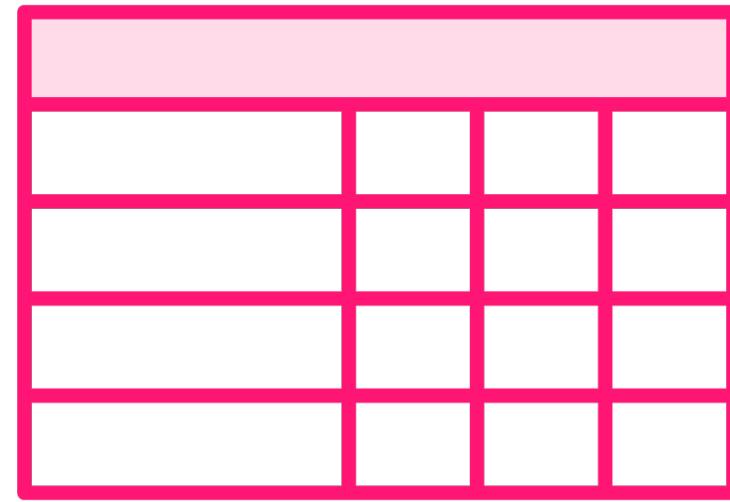
And our expression trees should be 100% independent



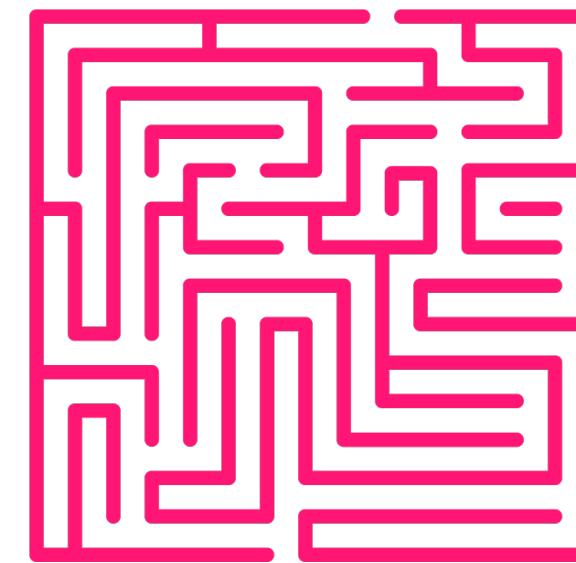


The Problem of Dynamic Filtering

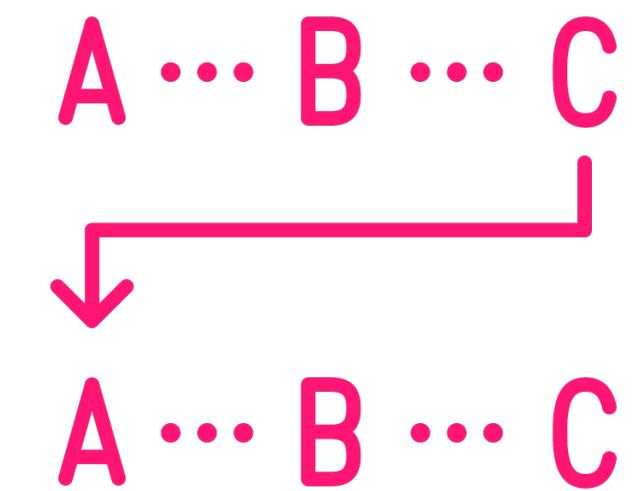
Implementing Filtering



Paste it into Excel



We *could* code
every path
through the code



Every possibility
multiplied together



A Better Approach

```
if(pClass != ""){  
    passengers = passengers.Where(passenger => passenger.PClass = pClass);  
}
```



Deferrable Execution

Enumerable will defer execution until the query must be materialized

Let's look at a purely expression-based approach



Demo



Review the filter interface

Take our request

Build a filter expression from it

Filter our data automagically





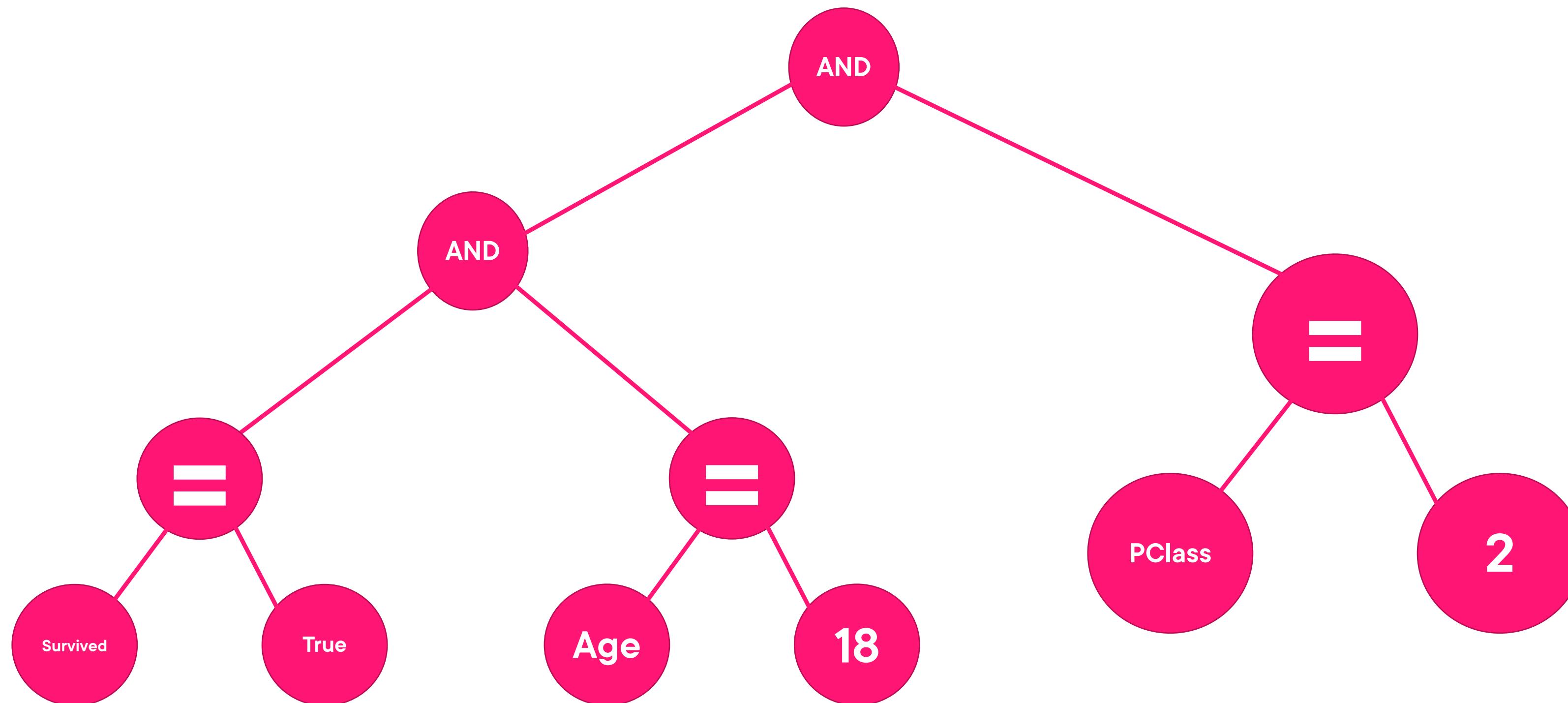
Closing the Loops



The Story So Far



Our Expression Tree



Demo



Replace a single where clause

With a function constructed from an expression tree

Apply that same logic to the others

We'll chain them all together with AND operators

Execute our expression tree



Demo



Look at a generic approach to what we were doing

A single function that generates and aggregates all of our expressions

Using a simple generic argument



Parsing a Lambda



Directions from Here

Implement a LINQ provider

That's a bunch
of work

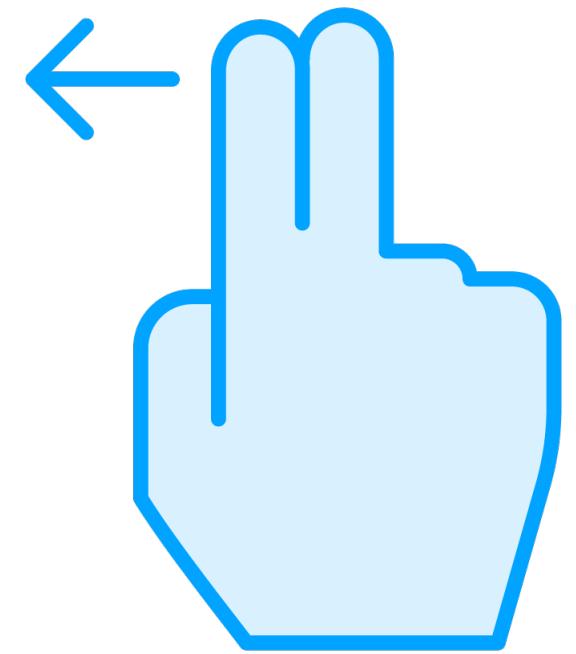
And you need to
understand the
target system

<https://bit.ly/3KX5o2p>

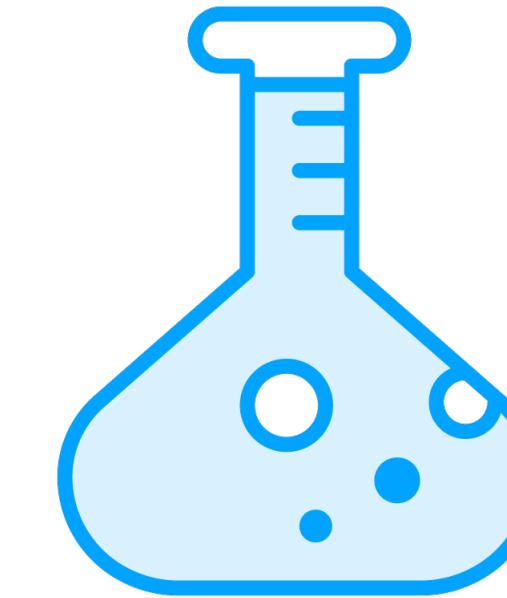
<https://bit.ly/37KJgta>



The Other Direction



**Let's go prior
to the expression**



**Generate an expression tree
from text**



The First Implementation

A simple expression parser

Put the query in the hands of
the user

Begin with the end in mind

The parser will need
a specific syntax



Demo



A package that makes our expressions more readable

We'll implement a quick expression parser

Text that the user enters in the interface



A Few Points About This

Ease of use vs. control

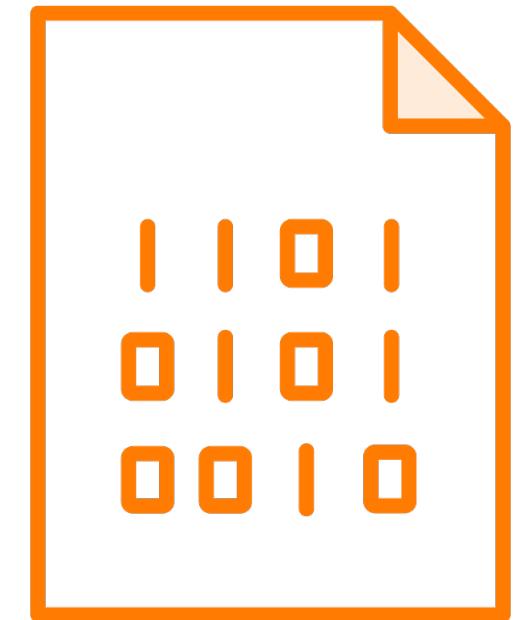
**There's a lot more
we could do**



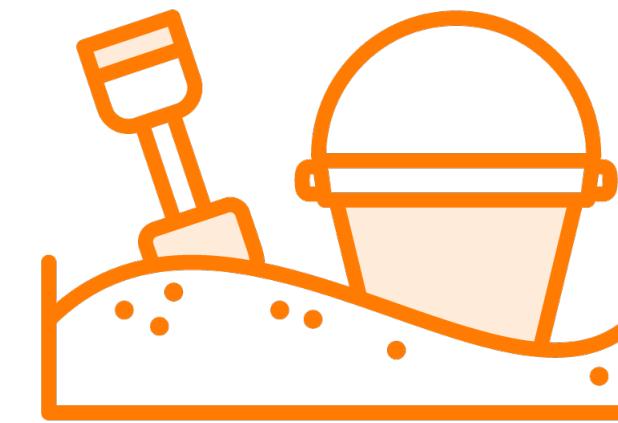
Towards a Scripting Engine with Expressions



Where We've Reached



Actual code from text



But we've stayed in
the Passenger sandbox



Creating an Expression for a Multi-line Function

We've kept it simple

No longer



Expressions are all there is.



The Function

```
if (number <= 1) return false;
if (number == 2) return true;
if (number % 2 == 0) return false;

var boundary = (int)Math.Floor(Math.Sqrt(number));

for (int i = 3; i <= boundary; i += 2)
    if (number % i == 0)
        return false;

return true;
```



Expressions are all there is.



Demo



Look at a unit test for our function

Take an in-depth look at the code

**That generates an expression tree that
detects the prime-ness of a number**

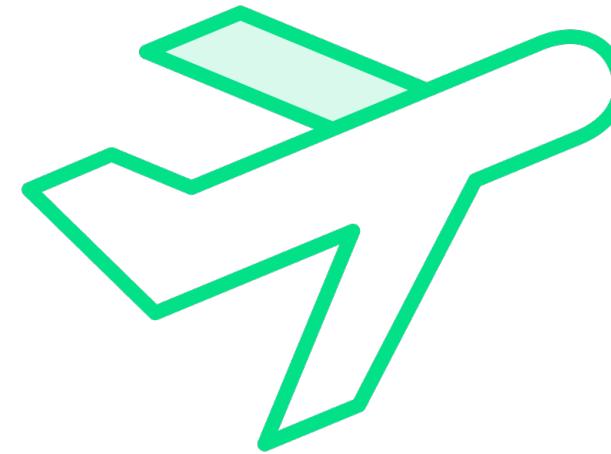
**Explore the different types of expressions
we can use to make that happen**



Working with Roslyn



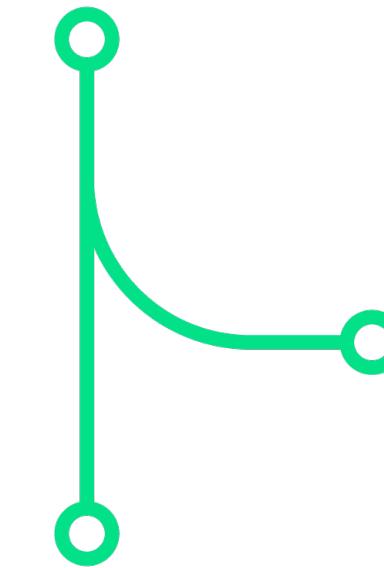
What is Roslyn?



**No conspiracy, just a
little town outside of
Seattle**



**A compiler for
VB and C#**



**An effort to
unify the work of
creating the compiler**



But Wait, There's More



Why should I care?

Because of the API

And because it's open source

<https://github.com/dotnet/roslyn>

The clone takes a WHILE

Create your own analyzer

Your own IDE

Runtime compilation



Credit Where Credit Is Due

[https://github.com/laurentkempe/
DynamicRun](https://github.com/laurentkempe/DynamicRun)

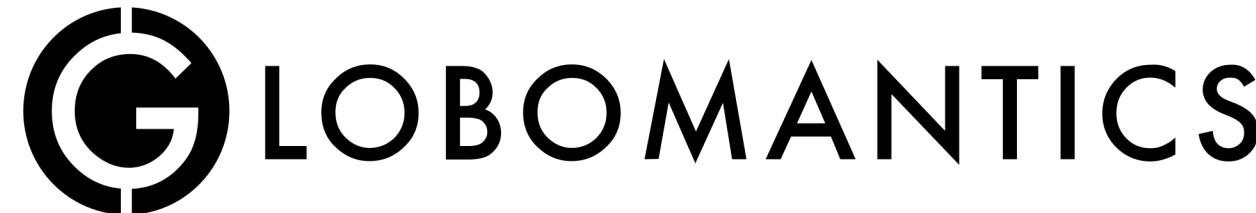
Microsoft MVP Laurent Kempè

I've tried to keep our
code close to his

<https://bit.ly/38IUWxC>



Globomantics Recruiting International



**Focused on matching opportunities
with candidates**

**Recruiters have their own perspectives on
what that means**

**The recruiters want the fine control that can
only come from script**

Making that happen is your job

**Yep, the recruiters will have to learn to code a
little bit**



Demo



A modified version of DynamicRun from Laurent Kempè

A process concept for enabling the recruiters' scripts

Execute the process

Make the script something more real

Look at the results



Midpoint Course Correction

That was the
hard part

We compile our
header and script,
and then execute

Now, let's take a
look at the script
itself



Making This Real

The code is
all yours

There is **no** sandbox

Run it in a
controlled container

Separate
compilation from
execution

Version control your
user's script files

Document your
script objects
to the user

<https://github.com/FeynmanFan/DynamicRun>



Course Summary



Delegates and Funcs

A bunch of LINQ

A bunch of Lambda

The Expression Types

A dynamic filter for the passengers of the Titanic

Made that filter text-driven

A pair of functions constructed manually via the expression libraries

A dynamic compiler platform

Truly running code at runtime



**THANK YOU VERY MUCH
FOR WATCHING!!!**

