Eliassen - LINQ Search Extensions

Summary

The intention of this library is to provide a common means to define endpoints for the searching, filtering, sorting and paging of data. As a mean to define a common interface and allow for faster development it was determined to create a common design pattern for all data query screens.

There was a preference to use existing off-the-shelf protocols and libraries as a means to provide this data querying functionality. Before designing this feature existing protocols were reviewed as such as OData and GraphQL. Various limitations such as complexity in implementation and potential exposure personally identifyable information it was determined an in-house solution was required.

Design Goals

- Create a query model that allows for an easy to follow and reusable pattern
- Data selection patterns
  - The dataset in particular will be provided via an HTTP/RESTful endpoint
  - Part of the URI path may be used to select the named endpoint
  - Search term may be used to select rows from multiple columns within the database
  - Filter rules may be used to select rows based on a particular field within the dataset
    - equal, not equal, wildcard (starts with, ends with, contains)
    - unbounded and bounded ranges
    - set value matching
- Multiple field sorting with the ability to select direction and priority order
- Data paging with the ability to change page size or disable paging to return all rows
- Make it easy to implement for developers

Technical Requirements

- Define Query Request Model
- Define Paged Result Models
- Define Patterns for