Supporting Sorting

Kevin Dockx

Architect

@KevinDockx https://www.kevindockx.com

Coming Up

Sorting resource collections

- One or more fields

- Ascending and descending

Sorting Collection Resources

Filtering

– http://host/api/authors?mainCategory=Singing

Searching

– http://host/api/authors?searchQuery=Kevin

Paging

– http://host/api/authors?pageNumber=1 &PageSize=5

```
http://host/api/authors?orderBy=age
```

Sorting Resource Collections

Outer Facing Model vs. Entity Model

Outer facing model (AuthorDto)

```
Guid Id

string FirstName

string LastName

int Age
```

Entity model (Author)

```
Guid Id

string FirstName

string LastName

DateTimeOffset DateOfBirth
```

Outer Facing Model vs. Entity Model

Outer facing model (AuthorDto)

```
Guid Id

string Name

int Age
```

Entity model (Author)

```
Guid Id

string FirstName

string LastName

DateTimeOffset DateOfBirth
```

Outer Facing Model vs. Entity Model

Outer facing model (AuthorDto)

Guid Id

string Name

int Age

float Royalties

Entity model (Author)

Guid Id

string FirstName

string LastName

DateTimeOffset DateOfBirth

Separating outer and entity models leads to more robust, reliable, and evolvable code

An important statement

```
http://host/api/authors?orderBy=age
```

Sorting Resource Collections

Resource fields, not entity (or other) fields

```
http://host/api/authors?orderBy=age desc
```

Sorting Resource Collections

Take direction into account

```
http://host/api/authors?orderBy=age desc,name
```

Sorting Resource Collections

Sort on multiple fields

Demo

Sorting collection resources

Creating a Property Mapping Service

A resource property might map to multiple properties on underlying objects

- Name maps to `FirstName`, `LastName`

A mapping might require reversing the sort order

- Sorting ascending by `Age` maps to sorting descending by `DateOfBirth`

Property Mapping Service Overview

PropertyMappingService : IPropertyMappingService

IList<IPropertyMapping> propertyMappings   eg: from AuthorDto to Author

PropertyMapping<TSource, TDestination> : IPropertyMapping

Dictionary<string, PropertyMappingValue>

PropertyMappingValue

DestinationProperties   eg: FirstName, LastName

Revert   eg: true for Age => DateOfBirth

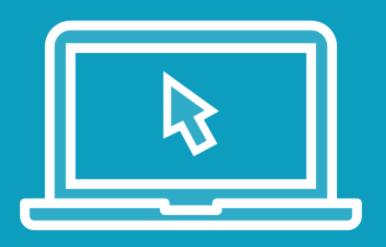**GetPropertyMapping <TSource, TDestination>()**   eg: from AuthorDto to Author

Demo

Creating a property mapping service

Demo

Creating a reusable ApplySort **method**

Demo

Adding `OrderBy` **clauses to pagination links**

Demo

Taking consumer errors into account when sorting

Summary

Collection resources are typically sorted on multiple fields

Take the sorting direction into account

Allow sorting on resource fields, not on fields that lie beneath that layer

Return the `OrderBy` **clause as query string in pagination links**

Errors should be returned as level 400 errors

Up Next:
Supporting Data Shaping