

Implementing Sorting and Data Shaping



Kevin Dockx

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up

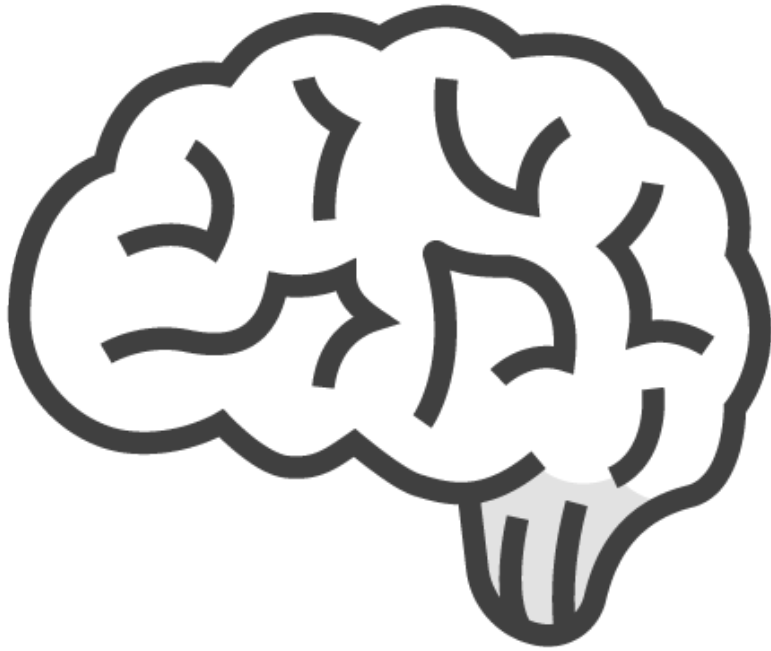


Sorting collection resources

Shaping resources (collection and single)



Sorting Collection Resources



<http://host/api/authors?orderBy=age>

- Resource fields

<http://host/api/authors?orderBy=age desc>

- Take direction into account

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

- Sort on multiple fields

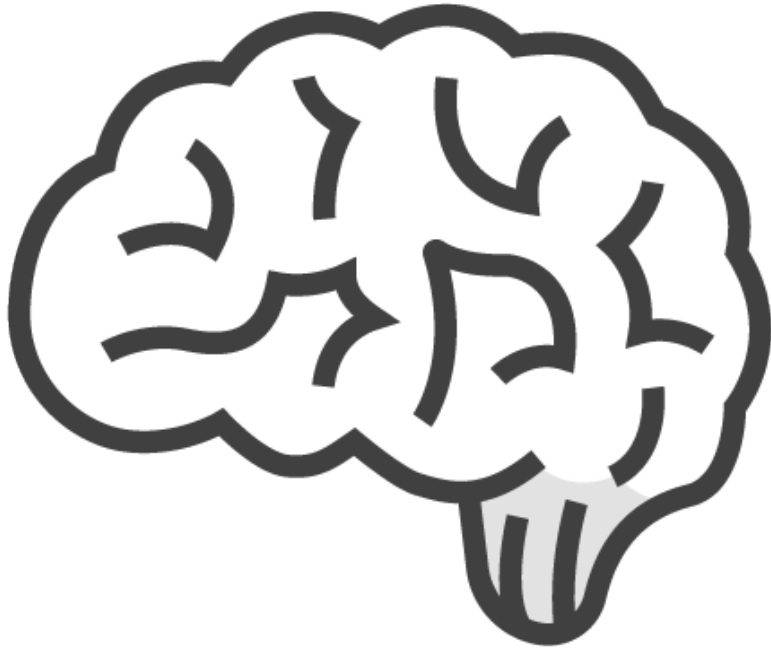
Demo



Sorting Collection Resources (Part 1)



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

ICollection<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



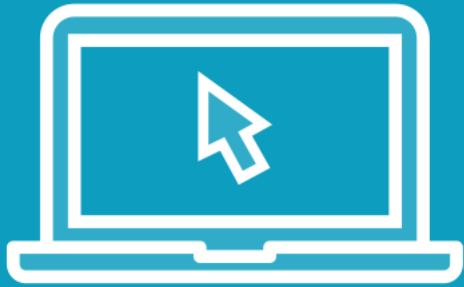
Demo



Creating a Property Mapping Service



Demo



Sorting Collection Resources (Part 2)



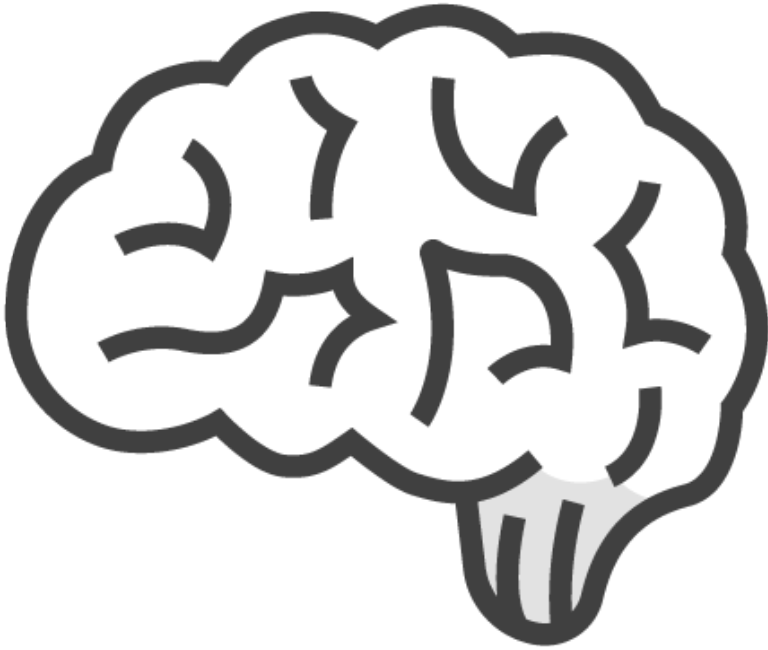
Demo



Taking Consumer Errors into Account When Sorting



Shaping Resources



Allows the consumer of the API to choose the resource fields

- <http://host/api/authors?fields=id,name>

Shape on fields of the resource, not on fields of lower-level layer objects



Demo



Creating a Reusable Extension Method to Shape Data



Demo



Shaping Collection Resources



Demo



Camel-casing Dictionaries When Serializing to JSON



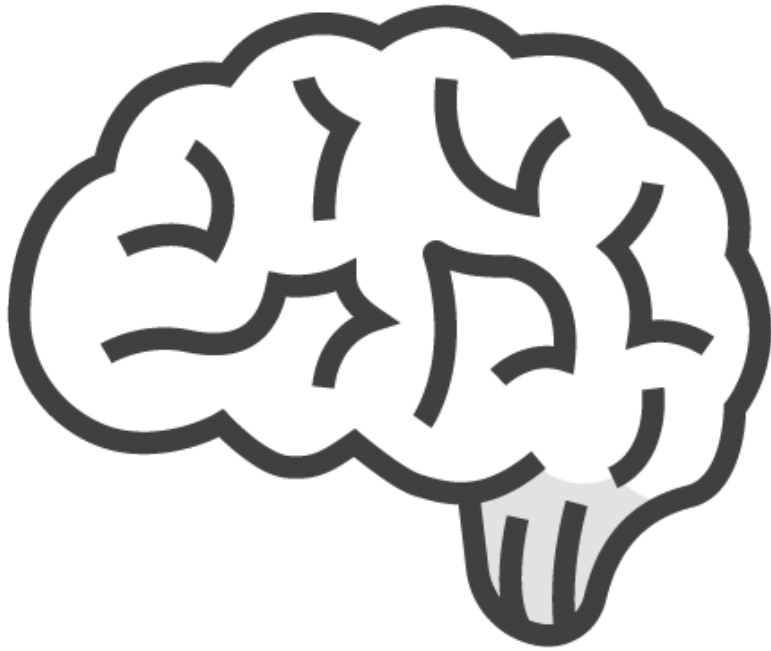
Demo



Shaping a Single Resource



Exploring Additional Options



Expand child resources

- <http://host/api/authors?expand=books>

Shape expanded resources

- <http://host/api/authors?fields=id,name,books.title>

Complex filters

- [http://host/api/authors?genre=contains\('Horror'\)](http://host/api/authors?genre=contains('Horror'))

Most APIs don't need all these options

Use query string parameters

Work on resource fields



Summary



Sorting

- Multiple fields
- Take direction into account

Data shaping

- Allow consumer to select resource fields

**Always work on fields of the resource,
not on fields of lower-level layer objects**

