

# Filtering and Searching

---



**Kevin Dockx**

ARCHITECT

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



# Coming Up



## Passing data to the API

- Binding source attributes
- ApiController

## Filtering and searching resources

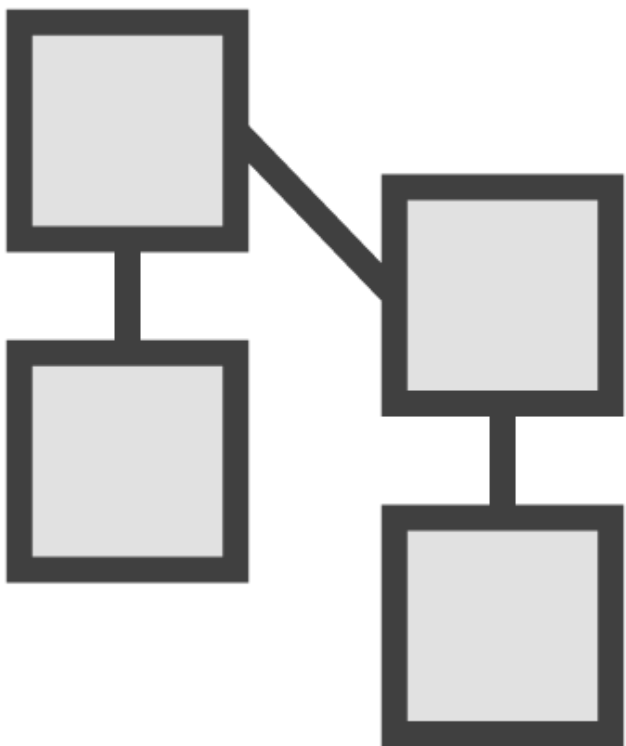


# Passing Data to the API

**Data can be passed to an API by various means**

**Binding source attributes tell the model binding engine where to find the binding source**





## **[FromBody]**

- Request body

## **[FromForm]**

- Form data in the request body

## **[FromHeader]**

- Request header

## **[FromQuery]**

- Query string parameters

## **[FromRoute]**

- Route data from the current request

## **[FromService]**

- The service injected as action parameter



```
public ActionResult<AuthorDto> GetAuthor(  
    [FromRoute] Guid authorId)
```

---

## Passing Data to the API

**Use binding source attributes to explicitly state where the action parameter should be bound from**



```
public ActionResult<AuthorDto> GetAuthor(  
    [FromRoute] Guid authorId)  
  
public ActionResult<IEnumerable<AuthorDto>> GetAuthors(  
    [FromQuery] string searchQuery)
```

---

## Passing Data to the API

Use binding source attributes to explicitly state where the action parameter should be bound from

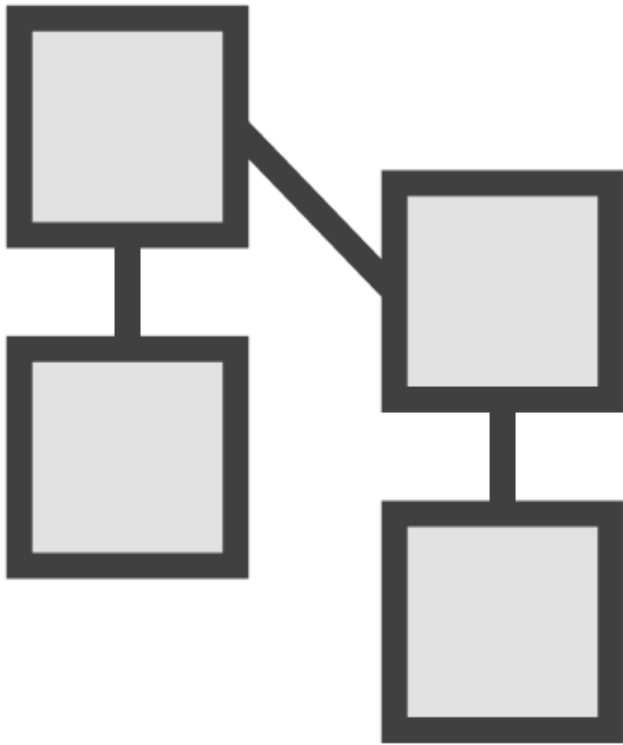


# Passing Data to the API

By default ASP.NET Core attempts to use the complex object model binder

The [ApiController] attribute changes the rules to better fit APIs





## [FromBody]

- Inferred for complex types

## [FromForm]

- Inferred for action parameters of type `IFormFile` and `IFormFileCollection`

## [FromRoute]

- Inferred for any action parameter name matching a parameter in the route template

## [FromQuery]

- Inferred for any other action parameters



# Filtering

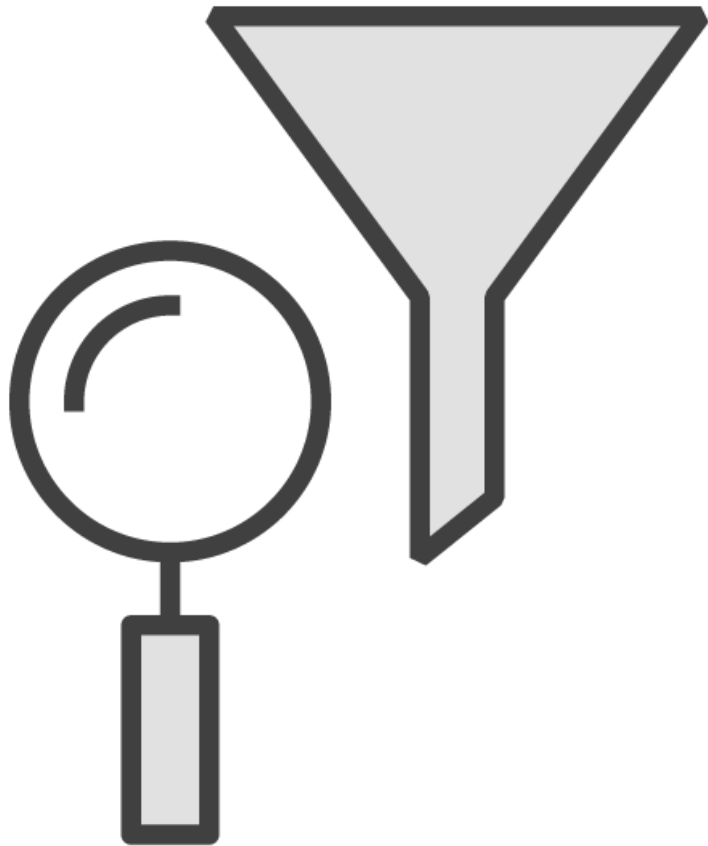
Filtering a collection means limiting the collection taking into account a predicate



# Searching

Searching a collection means adding matching items to the collection based on a predefined set of rules





Filtering allows you to be precise by adding filters until you get exactly the results you want

Searching allows you to go wider – it's used when you don't exactly know which items will be in the collection

`https://host/api/authors?maincategory=rum  
&searchQuery=pirate`

---

## Filtering and Searching

**Filter and search options are not part of the resource**

**Only allow filtering on fields that are part of the resource**



# Demo



## Filtering resource collections



# Demo



Searching through resource collections



# Deferred Execution

Query execution occurs sometime after the query is constructed



# Deferred Execution

**A query variable stores query commands, not results**

- IQueryable<T>: creates an expression tree

**Execution is deferred until the query is iterated over**

- foreach loop
- ToList(), ToArray(), ToDictionary()
- Singleton queries





# Demo



Grouping action parameters together into one object



# Summary



## [ApiController]

- Complex type parameters are inferred from the request body
- Action parameters of type `IFormFile` & `IFormFileCollection` are inferred as coming from a form
- Any action parameter name matching a parameter in the route template is inferred from the route
- Any other action parameters are inferred from the query string

# Summary



**When required, override default binding source behavior with binding source attributes**



# Summary



Filtering allows you to be precise by adding filters until you get exactly the results you want

Searching allows you to go wider – it's used when you don't exactly know which items will be in the collection



# Summary



**Deferred execution means query execution occurs sometime after the query is constructed**

