

# Manipulating Resources



**Kevin Dockx**

Architect

@Kevindockx | [www.kevindockx.com](http://www.kevindockx.com)

# Coming Up



## Creating a resource

- Generating links

## Updating a resource

## Deleting a resource

## Dealing with validation



# Routing, Revisited

HTTP method	Endpoint route builder method	Request payload	Sample URI	Response payload
GET	MapGet	-	/dishes /dishes/{dishId}	dish collection single dish
POST	MapPost	single dish	/dishes	single dish
PUT	MapPut	single dish	/dishes/{dishId}	single dish or empty
DELETE	MapDelete	-	/dishes/{dishId}	-



# Routing, Revisited

HTTP method	Endpoint route builder method	Request payload	Sample URI	Response payload
GET	MapGet	-	/dishes /dishes/{dishId}	dish collection single dish
POST	MapPost	single dish	/dishes	single dish
PUT	MapPut	single dish	/dishes/{dishId}	single dish or empty
DELETE	MapDelete	-	/dishes/{dishId}	-



# Routing, Revisited

HTTP method	Endpoint route builder method	Request payload	Sample URI	Response payload
GET	MapGet	-	/dishes /dishes/{dishId}	dish collection single dish
POST	MapPost	single dish	/dishes	single dish
PUT	MapPut	single dish	/dishes/{dishId}	single dish or empty
DELETE	MapDelete	-	/dishes/{dishId}	-



# Routing, Revisited

**Make sure the URLs make sense**

- Don't create a dish via /people

**Use nouns in URLs, not verbs**

**Don't mix plural and singular nouns**



Naming guidelines are not technical limitations, but they keep your API contract clean and predictable



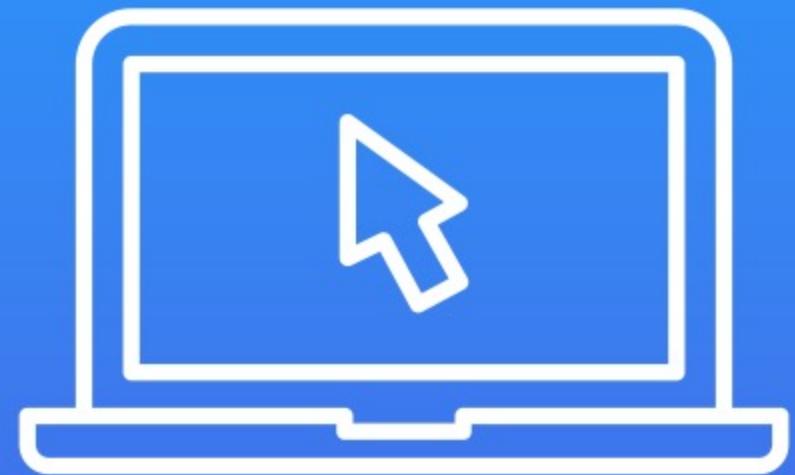
# Demo



## Creating a resource



# Demo



## Generating links



# Things to Keep in Mind When Creating a Resource



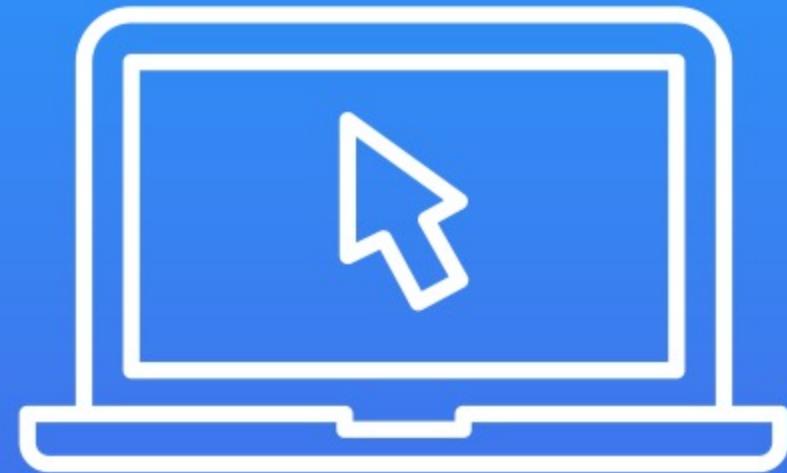
**When working with parent/child relationships, validate whether the parent exists**



**Don't use the same endpoint for creating one item & a collection of items. Create a new endpoint instead: [/itemcollections](#)**



# Demo



## Updating a resource



# Things to Keep in Mind When Updating a Resource



**Check if the resource exist, including hierarchical parents**



**Be careful when enabling PUT for collection resources: this can be very destructive**



**PUT is intended for FULL updates, PATCH is for partial updates**



# Things to Keep in Mind When Updating a Resource

Change sets for PATCH requests are often described as a list of operations: a `JsonPatchDocument`

- There is no support for this for minimal APIs...



# Demo



## Deleting a resource



# Demo



## Grouping resources



# Content negotiation

The process of selecting the best representation for a given response when there are multiple representations available

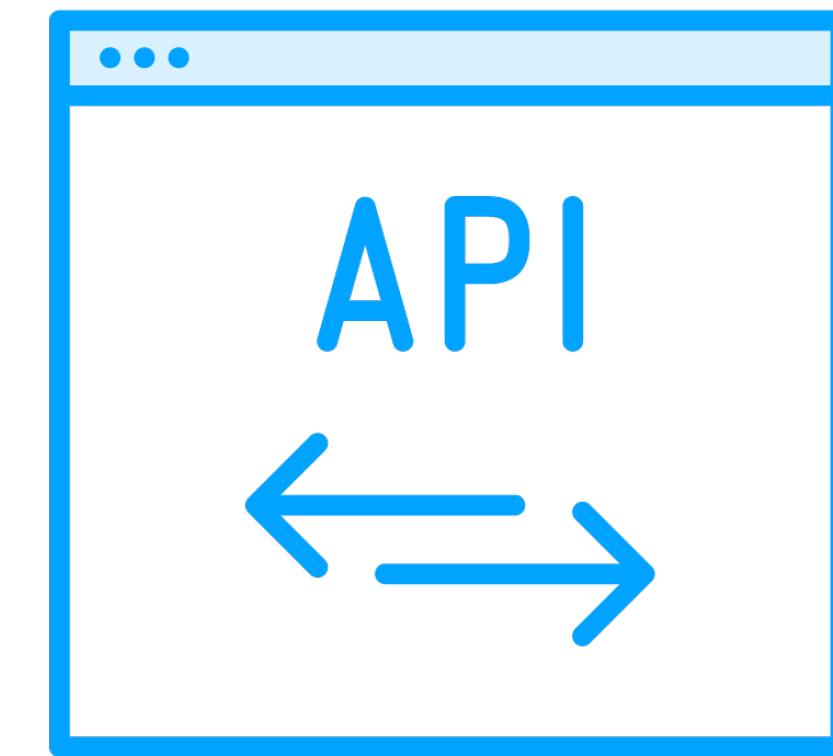


# Content Negotiation in Minimal APIs

GET api/dishes



{ dishes as JSON }

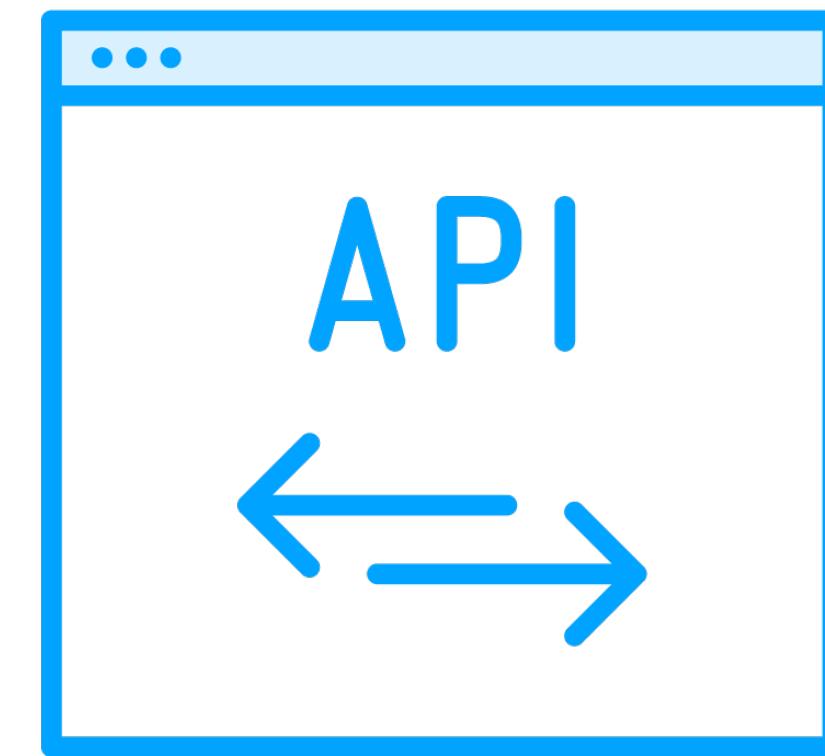


# Content Negotiation in Minimal APIs

GET api/dishes

Accept: application/json

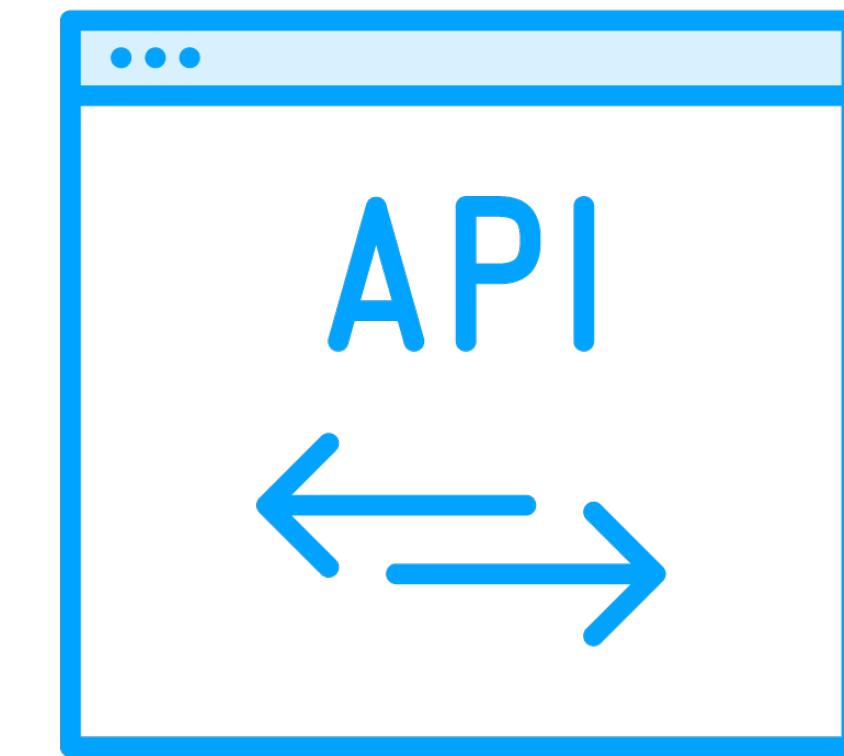
{ dishes as JSON }



# Content Negotiation in Minimal APIs

GET api/dishes

Accept: application/xml



{ dishes as XML }



# **Content Negotiation in Minimal APIs**

**Content negotiation is not supported out of  
the box by minimal APIs (nor is it planned)  
If you need it, consider ASP.NET Core MVC**



# Validation in Minimal APIs

**Input validation is a common requirement**

- Required fields, format, value & length restriction, cross-field rules, ...



```
public class DishForCreationDto
{
    [Required]
    [StringLength(100, MinimumLength = 3)]
    public required string Name { get; set; }
}
```

## A Common Approach to Validation: Annotations

### In ASP.NET Core MVC

- Automatic validation of incoming request bodies
- Manual validation with `(Try)ValidateModel(...)`.





**No built-in support for  
model validation for  
minimal APIs**

Nor is it planned...



# Summary



## Creating a resource

- MapPost

## Updating a resource

- MapPut

## Deleting a resource

- MapDelete



## Summary



**Group endpoints with `MapGroup` to apply common behavior to them**

**Generate links with the built-in `LinkGenerator`**

**Minimal APIs don't support content negotiation or validation out of the box**



**Up Next:**

# **Structuring Your Minimal API**

---

