

Gestión de recursos y validación de entradas

Resumen



Gestión de recursos

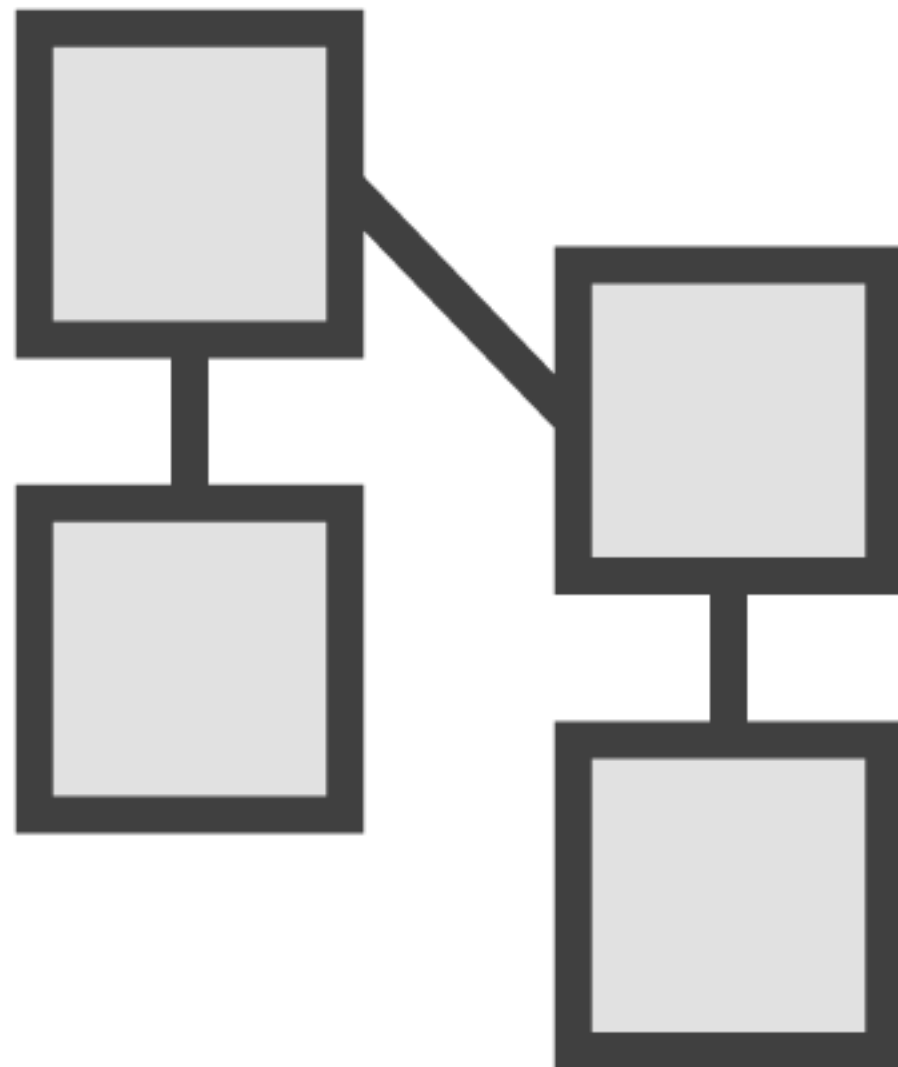
- Crear recursos
- Actualizar recursos
- Borrar recursos

Validar entrada

Pasar datos a la API

Los datos se pueden pasar a una API por varios medios

Enlazando los atributos de origen indican al motor de enlace de modelos dónde encontrar el origen del enlace



[FromBody]

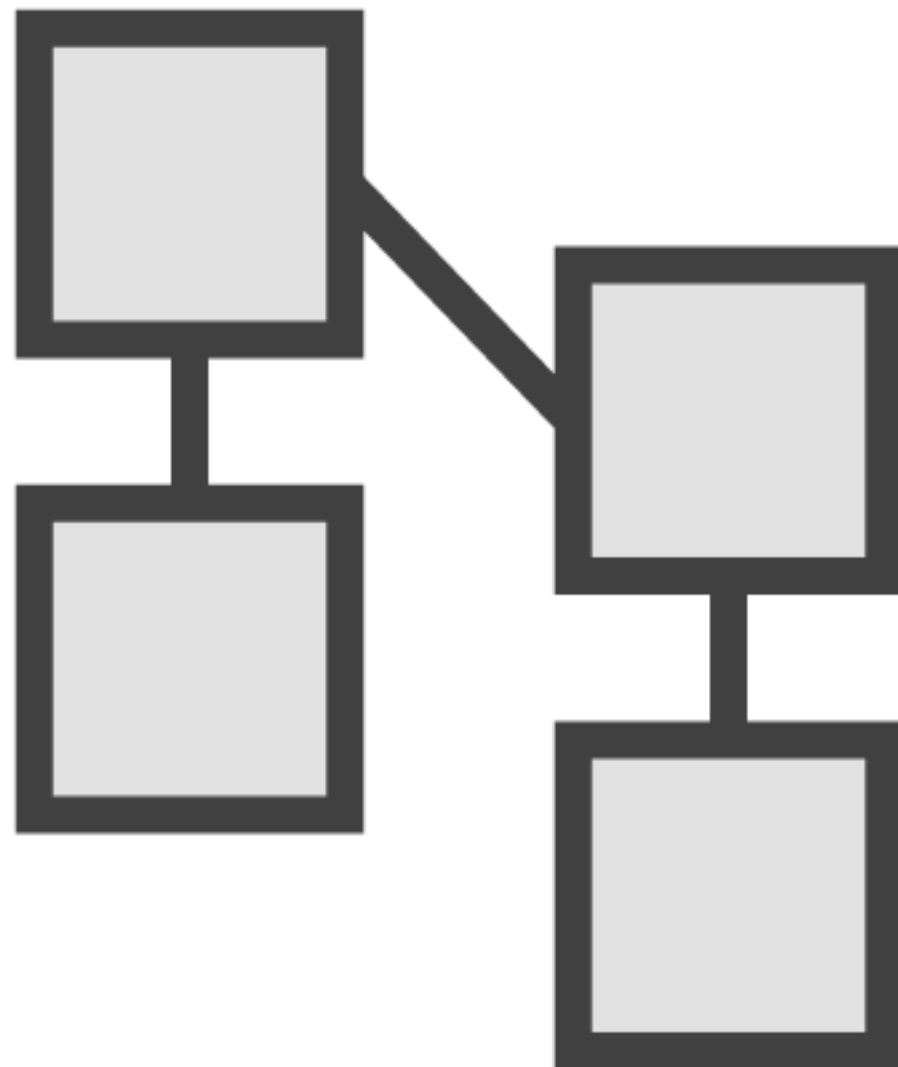
- Cuerpo de la petición

[FromForm]

- Datos del formulario en el cuerpo de la solicitud

[FromHeader]

- Cabecera de la petición



[FromQuery]

- Parámetros de la cadena de consulta

[FromRoute]

- Datos de ruta de la petición actual

[FromService]

- El servicio inyectado como parámetro de acción

```
public ActionResult<CityDto> GetCity(  
    [FromRoute] int cityId)
```

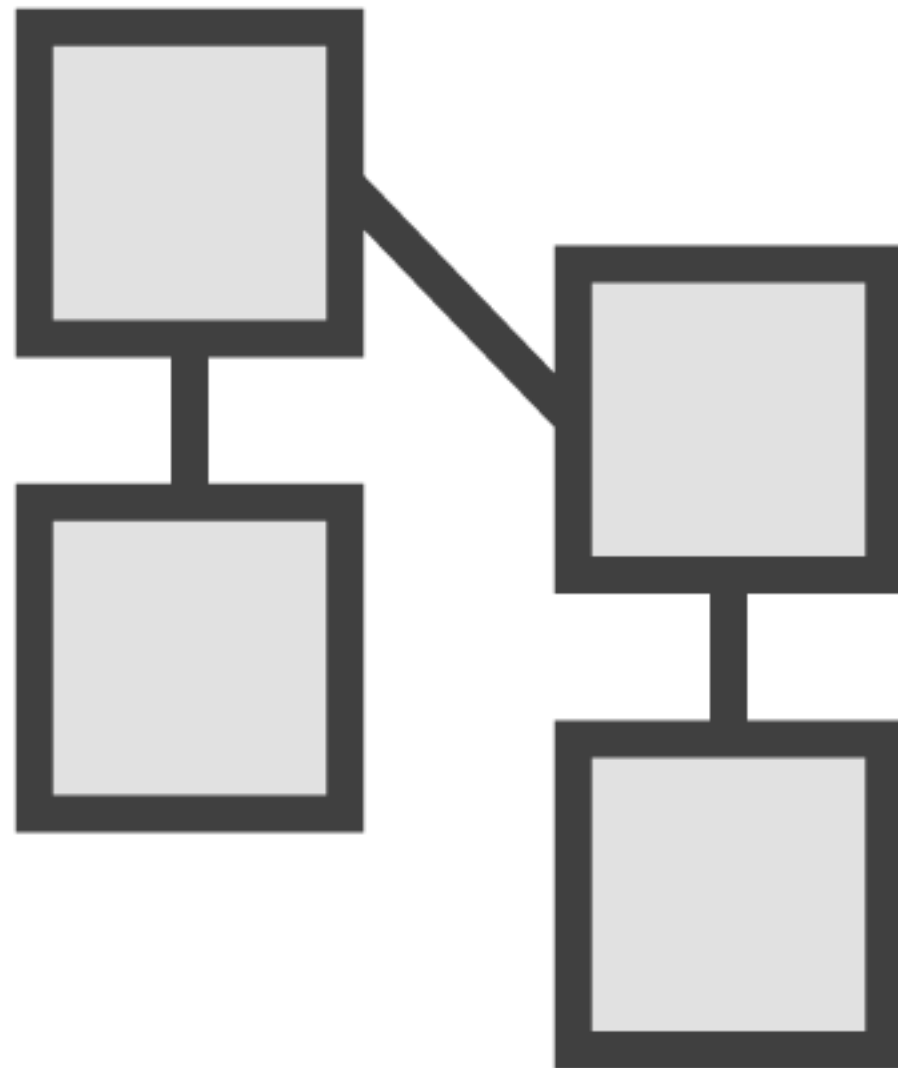
Pasando datos a la API

Utilice los atributos de origen de enlace para indicar explícitamente desde dónde se debe enlazar el parámetro de acción

Pasando datos a la API

Por defecto, ASP.NET Core intenta utilizar el enlazador modelo de objetos complejo

El atributo [ApiController] cambia las reglas para adaptarse mejor a las API

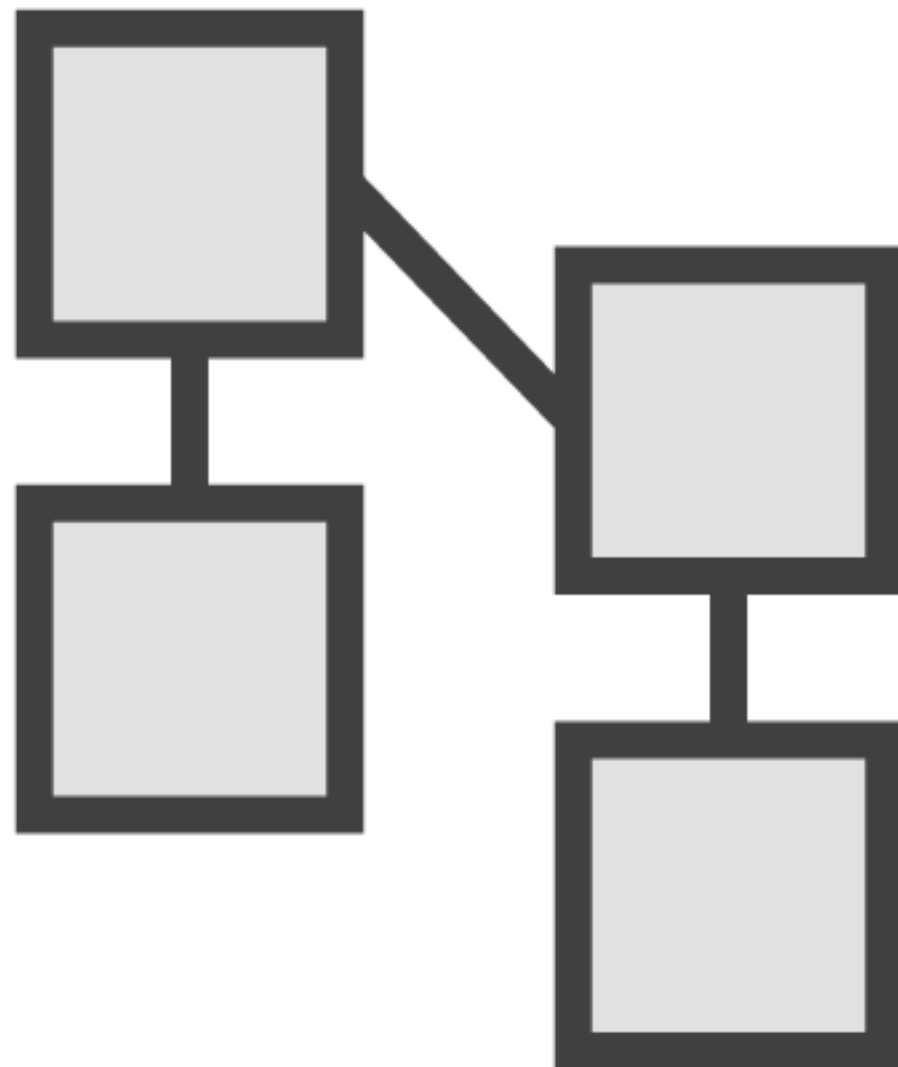


[FromBody]

- Inferido para tipos complejos

[FromForm]

- Inferido para parámetros de acción de tipo `IFormFile` y `IFormFileCollection`



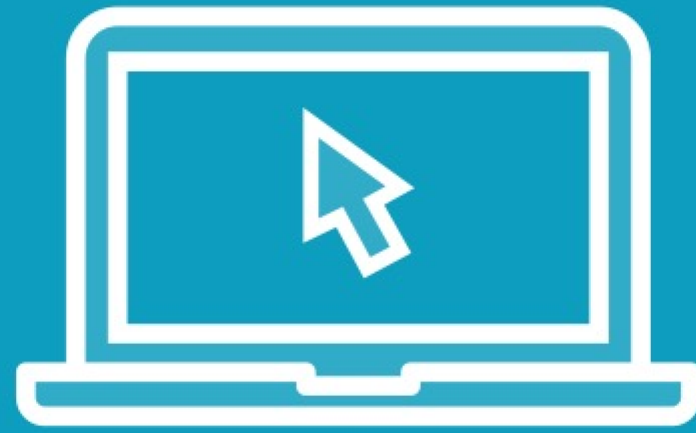
[FromRoute]

- Inferido para cualquier nombre de parámetro de acción que coincida con un parámetro en la plantilla de ruta

[FromQuery]

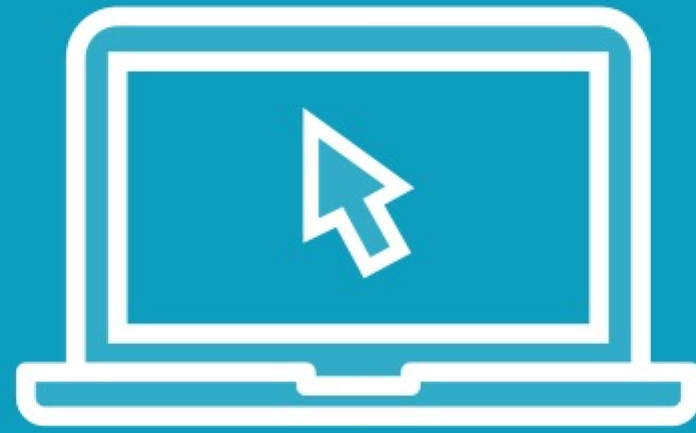
- Inferido para cualquier otro parámetro de la acción

Demo



Crear un recurso

Demo



Validar entrada

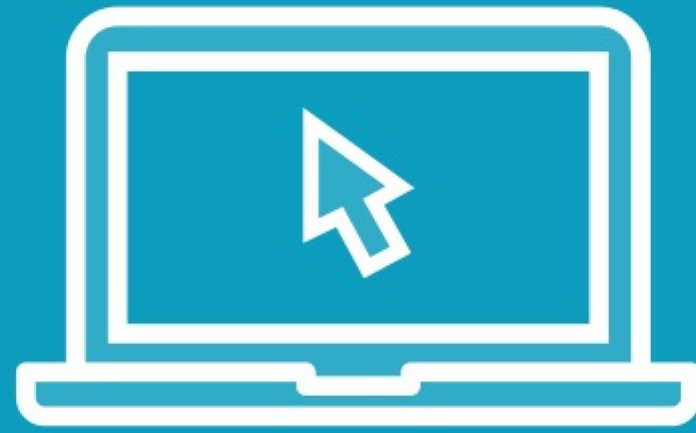
Una alternativa para validaciones

El enfoque integrado está bien para casos de uso sencillos

Para reglas complejas, puedes utilizar FluentValidation

– <https://fluentvalidation.net/>

Demo



Actualizar un recurso

Actualización parcial de un recurso

Json Patch (RFC 6902)

– <https://tools.ietf.org/html/rfc6902>

Describe una estructura de documento para expresar una secuencia de operaciones a aplicar a un documento JSON

```
[
  {
    "op": "replace",
    "path": "/name",
    "value": "new name"
  },
  {
    "op": "replace",
    "path": "/description",
    "value": "new description"
  }
]
```

◀ array de operaciones

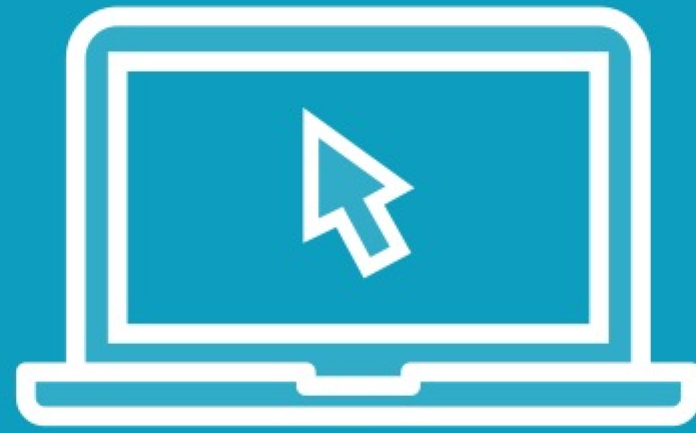
◀ operación “replace”

◀ la propiedad “name” cambia su valor a “new name”

◀ operación “replace”

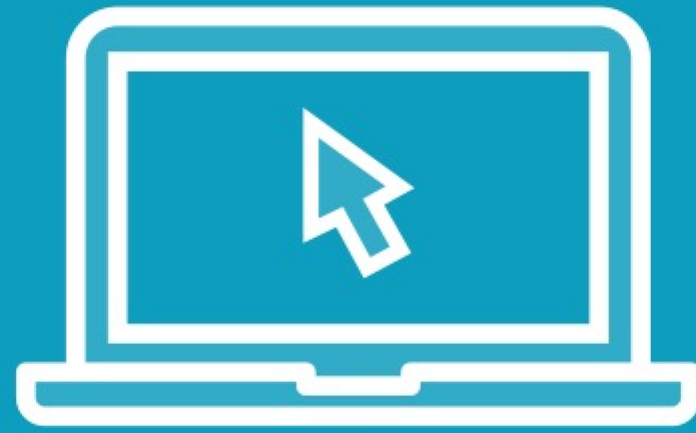
◀ la propiedad “description” cambia su valor a “new

Demo



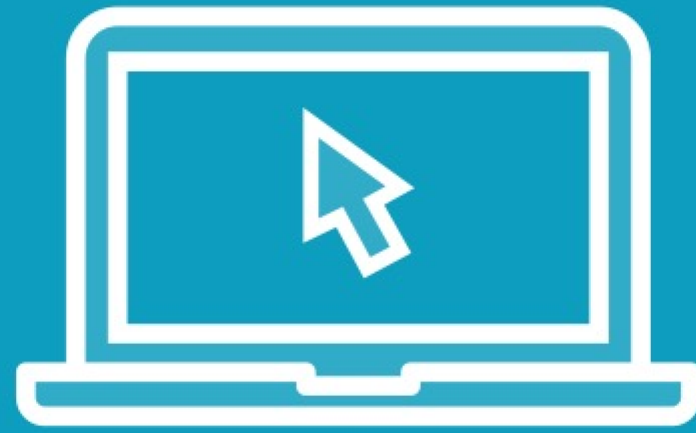
Añadir soporte de JsonPatchDocument

Demo



Actualización parcial de un recurso

Demo



Borrar un recurso

Resumen



Utiliza POST para crear un recurso

- 201 Created
- Header: Content-Type

Validación

- Data annotations
- ModelState

Resumen



**Usa PUT para actualizaciones completas,
PATCH para actualizaciones parciales**

- JsonPatch standard
- 204 NoContent or 200 Ok

DELETE para borrar recursos

- 204 NoContent

A continuacion:

Servicios e inyección de dependencias
