

Sesión 03

Estilo

Arquitectónico
REST

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe



8 NET

FULL-STACK
DEVELOPER

ÍNDICE

01 Parámetros de Filtrado y Búsqueda

02 Ordenamiento de resultados

03 Paginación de datos de respuesta

04 Estructura de datos de respuesta dinámica

05 HATEOAS y Negociación de contenido

01

Parámetros de Filtrado y Búsqueda

Filtrado

```
https://localhost/api/authors?mainCategory=Singing
```

Filtrar una colección significa limitar la colección teniendo en cuenta un predicado

Pasar el nombre y el valor del campo a través del Query String

Búsqueda

```
https://localhost/api/authors?searchQuery=pirate
```

Buscar una colección significa agregar elementos coincidentes a la colección en función de un conjunto predefinido de reglas.

Pasar texto para buscar a través del Query String

El filtrado le permite ser preciso agregando filtros hasta que obtenga exactamente los resultados que desea

La búsqueda le permite ir más allá: se usa cuando no sabe exactamente qué artículos estarán en la colección.

Filtrado y búsqueda

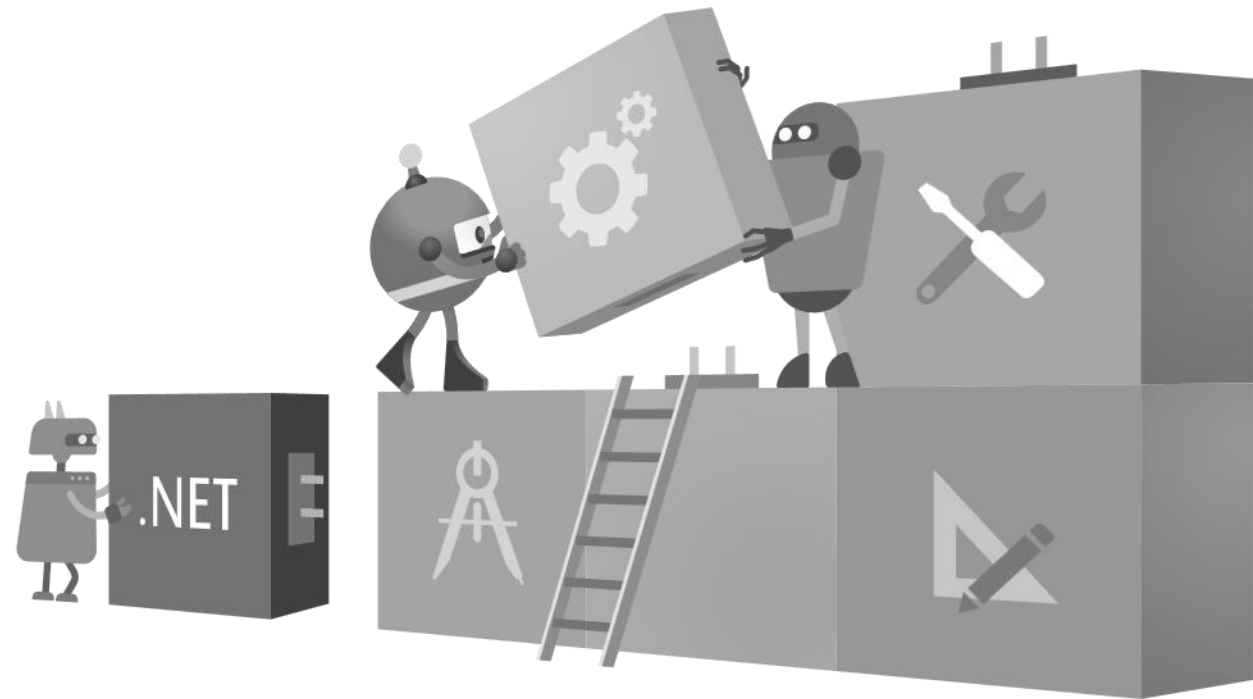
```
https://localhost/api/authors?mainCategory=Singing&searchQuery=pirate
```

Las opciones de filtro y búsqueda **no forman parte del recurso.**

Permitir únicamente el filtrado en los campos que forman parte del recurso

Parámetros de Filtrado y Búsqueda

DEMO



03



Ordenamiento de resultados

Ordenar colecciones de recursos



Campos de recursos, no campos de entidad (u otros)

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

Ordenar colecciones de recursos



Tener en cuenta la dirección

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

Ordenar colecciones de recursos

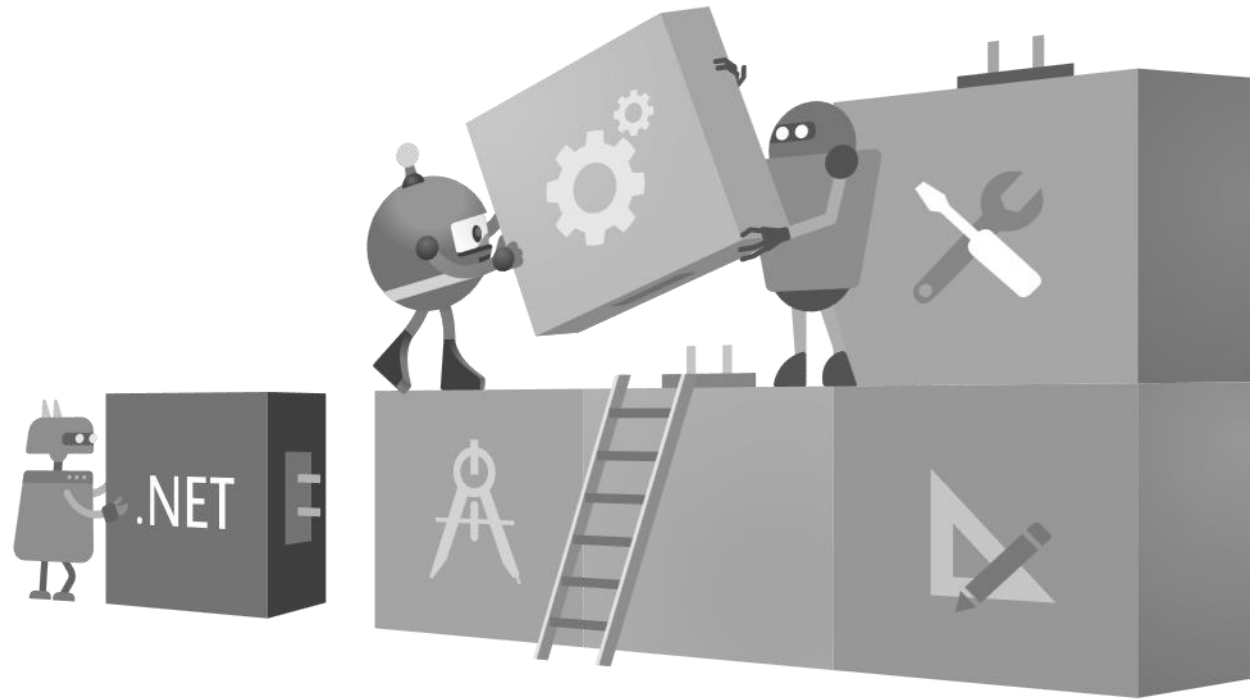


Ordenar en varios campos

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

Ordenamiento de resultados

DEMO

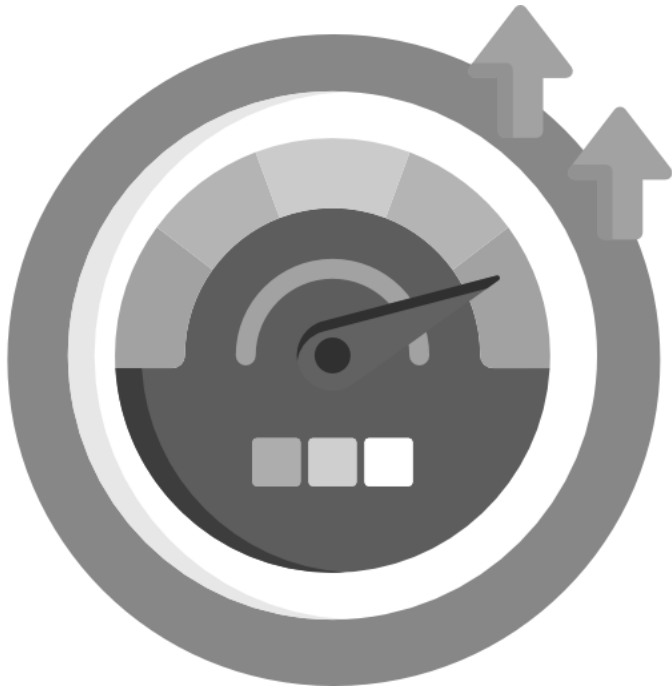


02



Paginación de datos de respuesta

Paginación a través de Recursos de colección

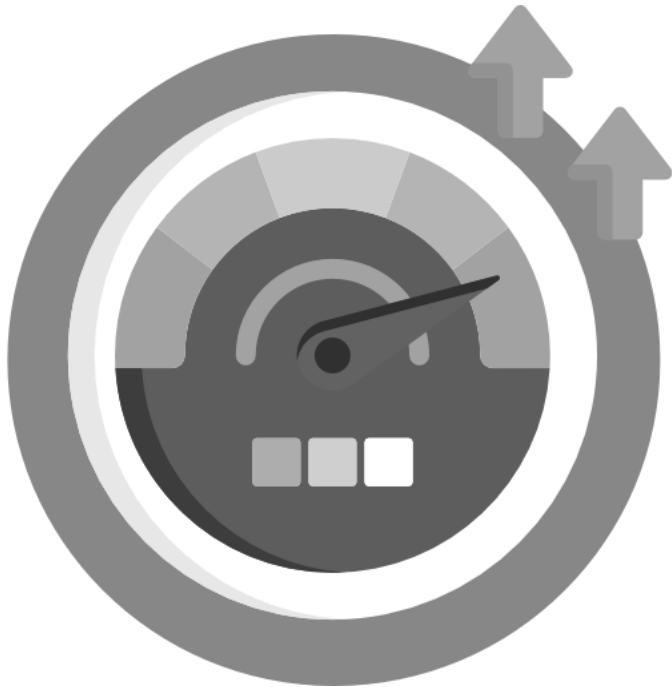


Los recursos de colección a menudo crecen bastante grandes

- Implementar paginación en todos ellos

La paginación ayuda a evitar problemas de rendimiento

Paginación a través de Recursos de colección



Los parámetros se pasan a través de Query String

- <http://localhost/api/authors?pageNumber=1&pageSize=5>

Paginación a través de Recursos de colección



Limitar el tamaño máximo de página

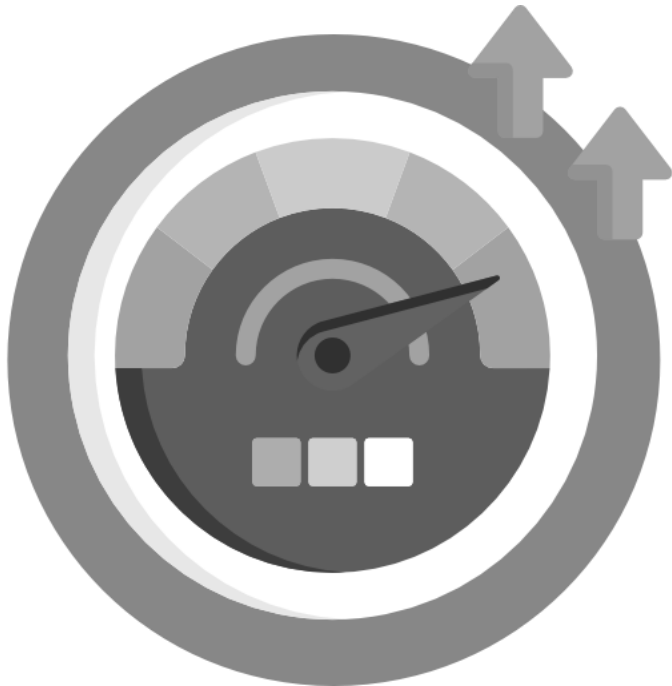


Página por defecto



Página hasta el almacén de datos

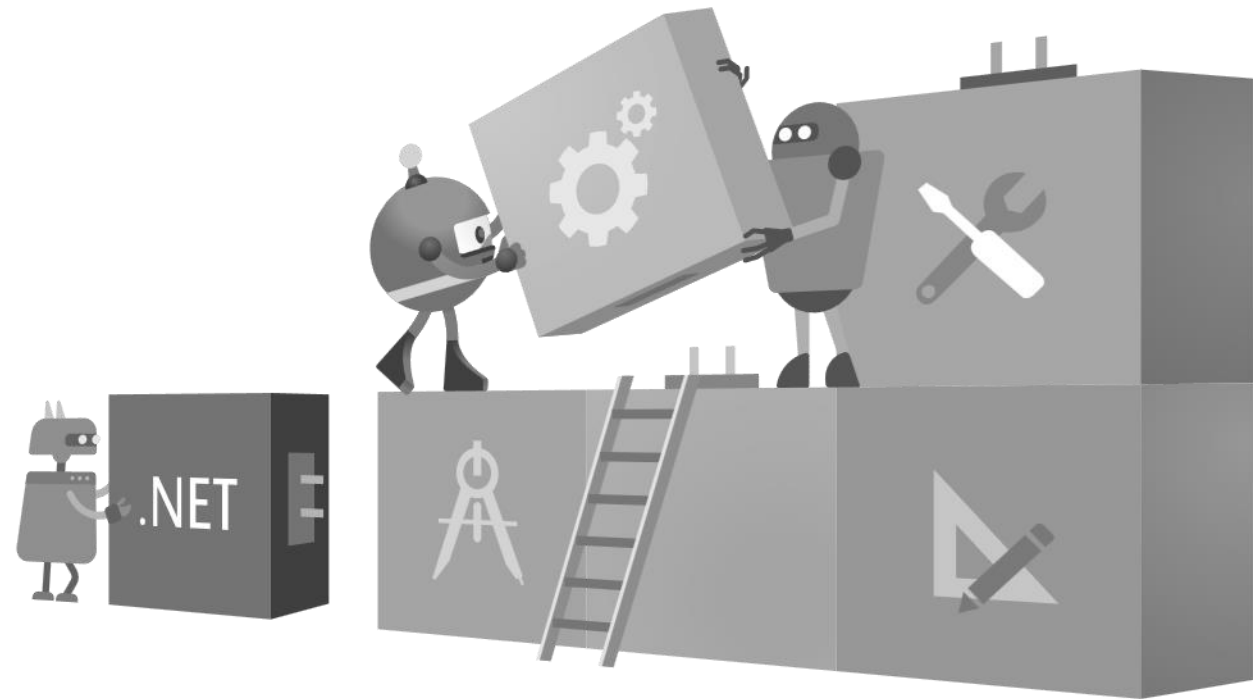
Paginación a través de Recursos de colección



- Debe incluir al menos enlaces a las páginas anterior y siguiente.
- Podría incluir información adicional: recuento total, cantidad de páginas, ...
- Al solicitar application/json, los metadatos de paginación no forman parte de la representación de recursos
- **Usar un encabezado personalizado, como X-Pagination**

Paginación de datos de respuesta

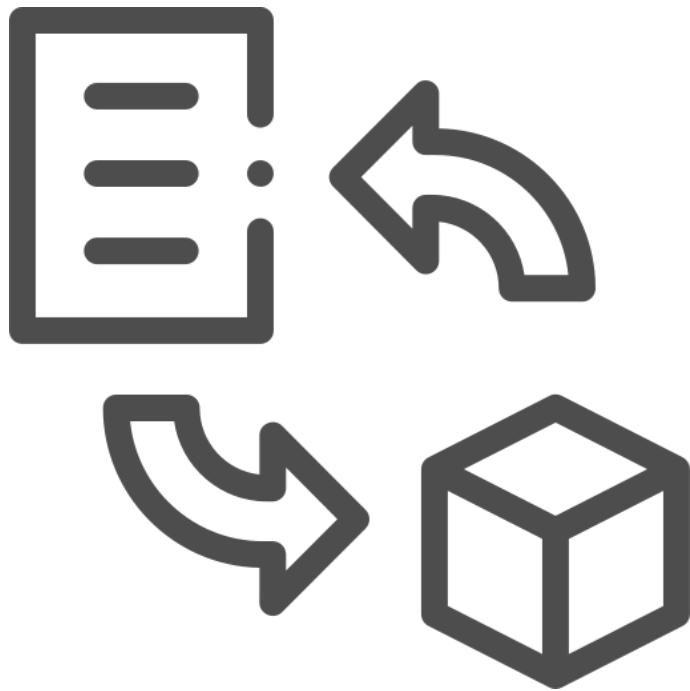
DEMO



04

Estructura de datos de respuesta dinámica

Configuración de datos (Shaping) de Recursos



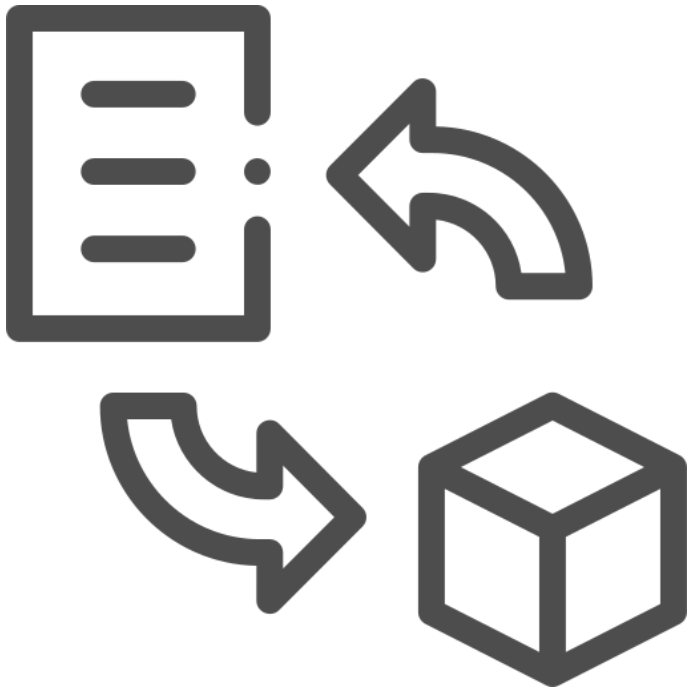
El Shape de datos permite al consumidor de la API elegir los campos de recursos.

El Shape es a los campos del recurso, no a los campos de objetos de las capas de nivel inferior

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

→ Estructura de datos de respuesta dinámica

Configuración de datos (Shaping) de Recursos



Exploración de opciones adicionales

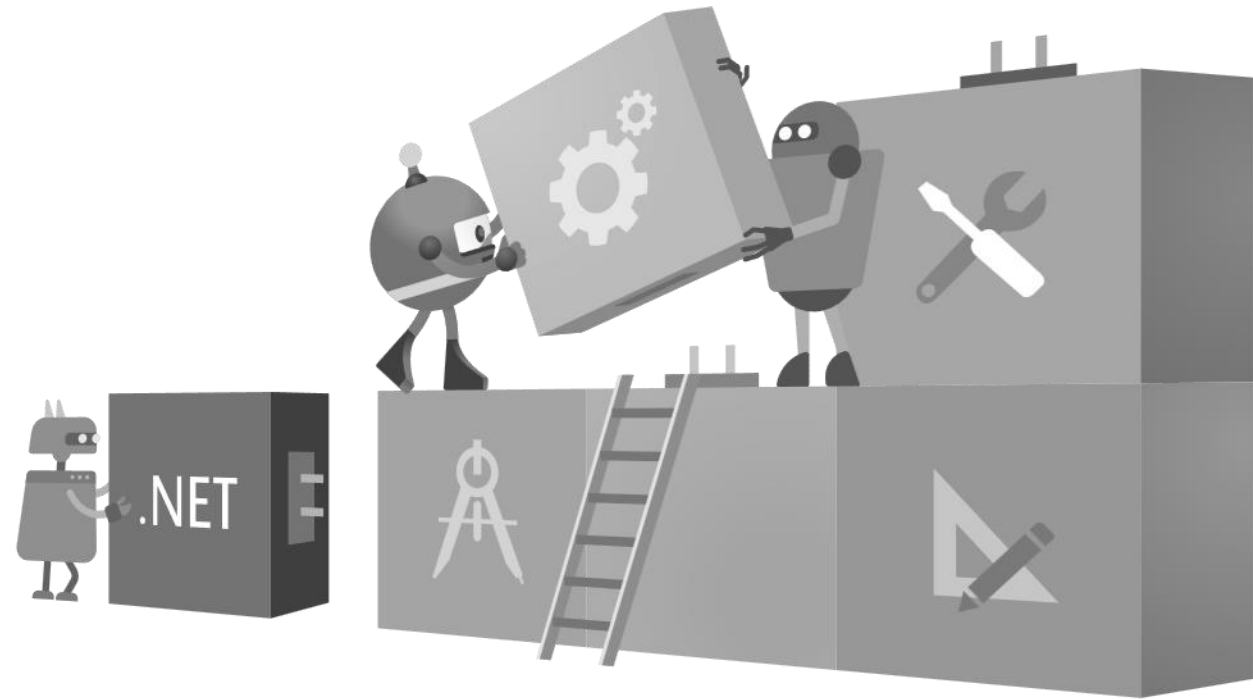
Incluir recursos hijos

<http://localhost/api/authors?expand=course>

<http://localhost/api/authors?fields=courses.id>

Estructura de datos de respuesta dinámica

DEMO



05

HATEOAS y Negociación de contenido

El hipermedia como motor de estado de aplicación



Ayuda con la capacidad de evolución y la autodescriptividad

Hypermedia impulsa cómo consumir y usar la API.

HTML representa enlaces con el elemento anchor

- **href:** Contiene el URI
- **rel:** Describe cómo se relaciona el vínculo con el recurso
- **type:** describe el media type

HATEOAS

```
{ ...  
  "numberOfAvailablePlaces": 10,  
  "content": "mature",
```

HATEOAS

```
{ ...  
  "numberOfAvailablePlaces": 10,  
  "content": "mature",  
  "links":
```

HATEOAS

```
{ ...  
  "numberOfAvailablePlaces": 10,  
  "content": "mature",  
  "links": [  
    {  
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",  
      "rel": "self",  
      "method": "GET"  
    },  
  ],  
}
```

HATEOAS

```
{ ...
  "numberOfAvailablePlaces": 10,
  "content": "mature",
  "links": [
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "update-course-full",
      "method": "PUT"
    },
  ],
}
```

HATEOAS

```
{ ...  
  "numberOfAvailablePlaces": 10,  
  "content": "mature",  
  "links": [  
    {  
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",  
      "rel": "self",  
      "method": "GET"  
    },  
    {  
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",  
      "rel": "update-course-full",  
      "method": "PUT"  
    },  
    {  
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",  
      "rel": "update-course-partial",  
      "method": "PATCH"  
    }  
  ],  
}
```

HATEOAS

```
{ ...
  "numberOfAvailablePlaces": 10,
  "content": "mature",
  "links": [
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "update-course-full",
      "method": "PUT"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "update-course-partial",
      "method": "PATCH"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "delete-course",
      "method": "DELETE"
    }
  ],
}
```

HATEOAS

```
{ ...
  "numberOfAvailablePlaces": 10,
  "content": "mature",
  "links": [
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "update-course-full",
      "method": "PUT"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "update-course-partial",
      "method": "PATCH"
    },
    {
      "href": "http://localhost/api/authors/{authorId}/courses/{courseId}",
      "rel": "delete-course",
      "method": "DELETE"
    },
    {
      "href": "http://localhost/api/coursereservations",
      "rel": "reserve-course",
      "method": "POST"
    }
  ]
}
```


Uso de HATEOAS para enlaces de paginación

```
{ ...
  "links": [ ...,
    {
      "href": "http://host/api/authors?pageNumber=1&pageSize=10",
      "rel": "previous-page",
      "method": "GET"
    },
    {
      "href": "http://host/api/authors?pageNumber=3&pageSize=10",
      "rel": "next-page",
      "method": "GET"
    }
  ]
}
```

Negociación de contenido

"Una API REST debe dedicar casi todo su esfuerzo descriptivo a definir los media types utilizados para representar recursos y controlar el estado de la aplicación, o en definir nombres de relaciones extendidas y/o hypertext-enabled mark-up para los media types estándar existentes".

Roy Fielding
<https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>

Negociación de contenido

Media Types específicos del proveedor

Top-level type
↓
application/vnd.erick.hateoas+json

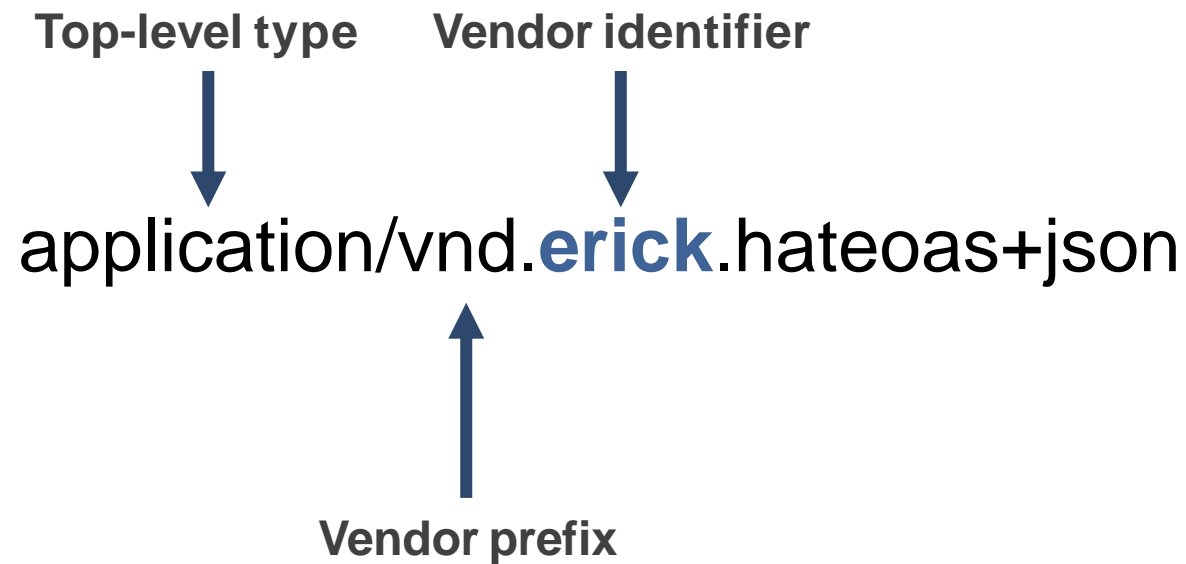
Negociación de contenido

Media Types específicos del proveedor



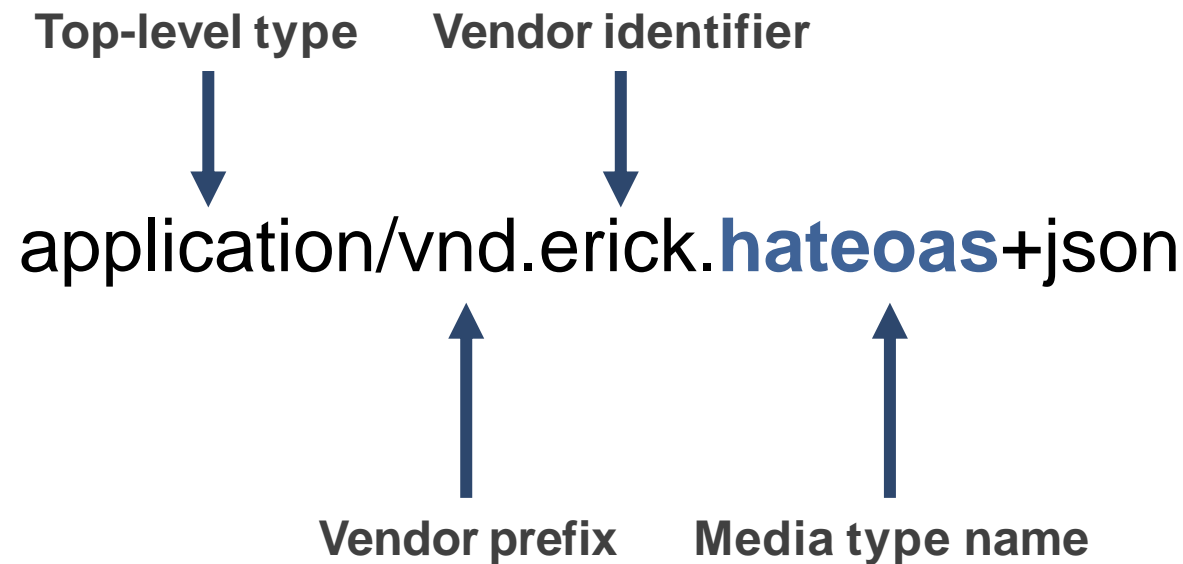
Negociación de contenido

Media Types específicos del proveedor



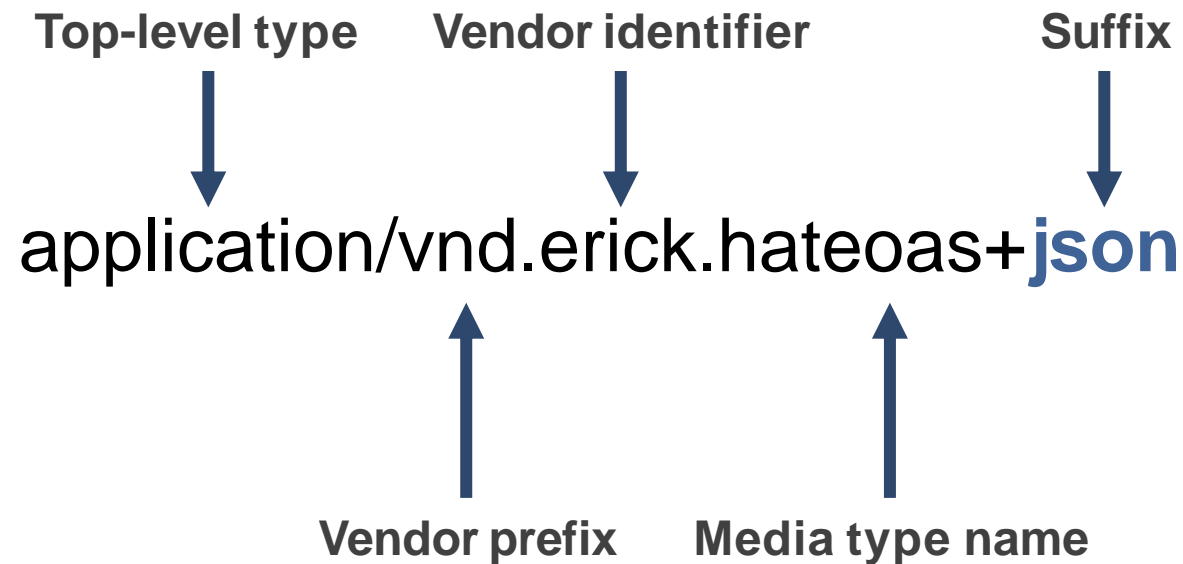
Negociación de contenido

Media Types específicos del proveedor



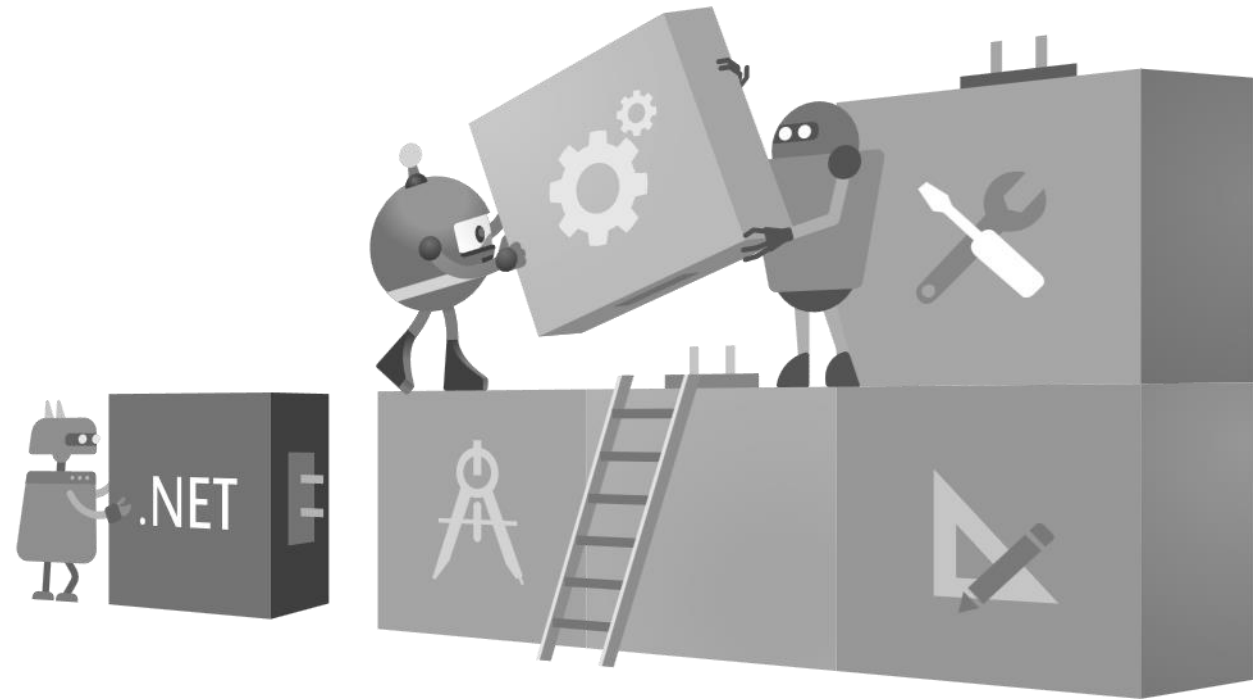
Negociación de contenido

Media Types específicos del proveedor



HATEOAS y Negociación de contenido

DEMO





GRACIAS
POR SU PREFERENCIA

