

# Advanced HATEOAS, Media Types, and Versioning

---



**Kevin Dockx**

ARCHITECT

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



# Coming Up



**HATEOAS and content negotiation**

**Expanding on media types, and a few words on versioning**



```
{ "value": [ {author}, { author} ],  
  "links": [ ...,  
    {  
      "href": "http://host/api/authors",  
      "rel": "self",  
      "method": "GET" }, ...]  
}
```

~~Accept: application/json?~~  
Accept: application/...

---

## HATEOAS and Content Negotiation

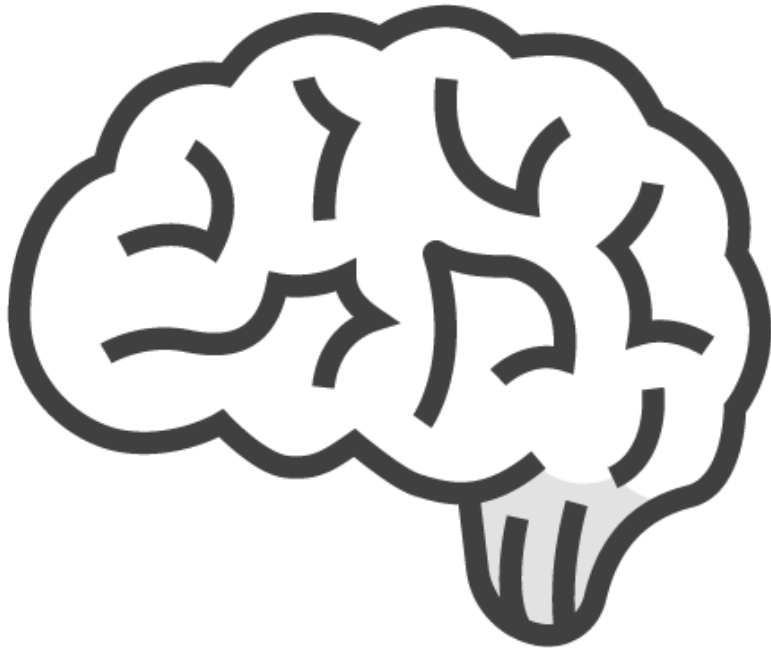
Metadata goes in the header

Explicit parts of the response go in the body

We're dealing with two different representations of the same resource



# HATEOAS and Content Negotiation



## Vendor-specific media type

- `application/vnd.marvin.hateoas+json`
- Vendor prefix, vendor identifier (, other), format

To get the links in the representation, we should request this media type



Demo



## HATEOAS and Content Negotiation



# Demo



## Working Towards Self-discoverability with a Root Document



```
{ "id": "bc4c35c3-3857-4250-9449-155fcf5109ec",  
  "title": "The Winds of Winter", ...  
  "links": [ ...,  
    {  
      "href": "http://host/api/authors/{authorId}/books/{bookId}",  
      "rel": "update-book-full",  
      "method": "PUT"  
    }  
  ]  
}
```

---

## Revisiting Media Types

**The link format is custom**

**We don't know what to send in the request body**



# Revisiting Media Types

## Author (GET)

```
{  
  "id": "76053df4-6687-4353-  
        8937-b45556748abe",  
  "name": "George RR Martin",  
  "age": 68,  
  "genre": "Fantasy"  
}
```

~~application/json~~

application/vnd.marvin.  
author.friendly+json

## Author (POST)

```
{  
  "firstName" : "George RR",  
  "lastName" : "Martin",  
  "dateOfBirth" : "1948-20-  
                  09T00:00:00",  
  "genre": "Fantasy"  
}
```

~~application/json~~

application/vnd.marvin.  
author.full+json





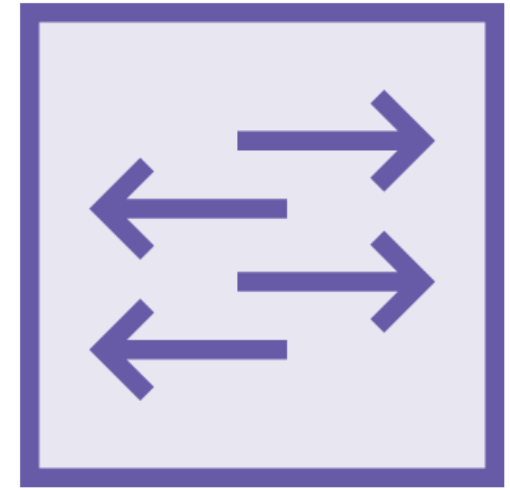
# Versioning in a RESTful World



Functionality

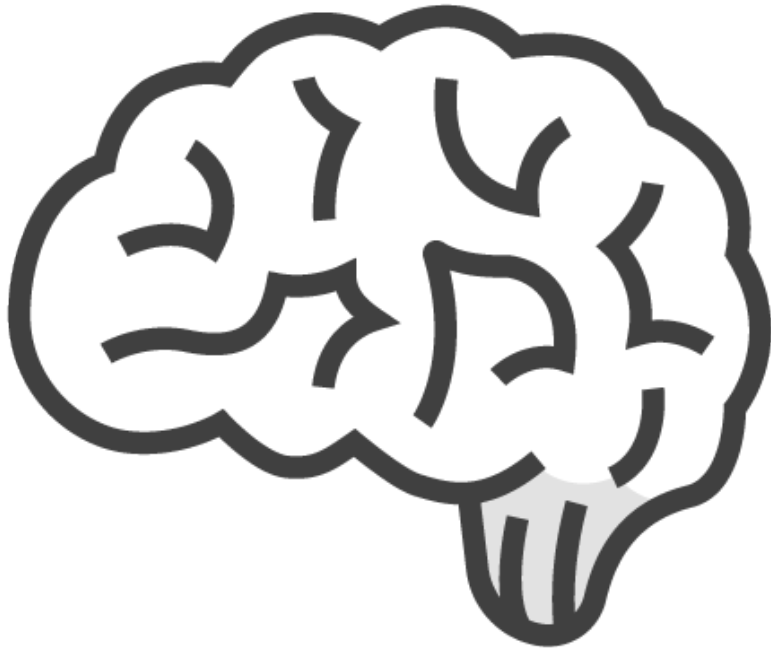


Business rules



Resource  
representations

# Versioning in a RESTful World



## Through the URI

- `api/v1/authors`

## Through query string parameters

- `api/authors?api-version=v1`

## Through a custom header

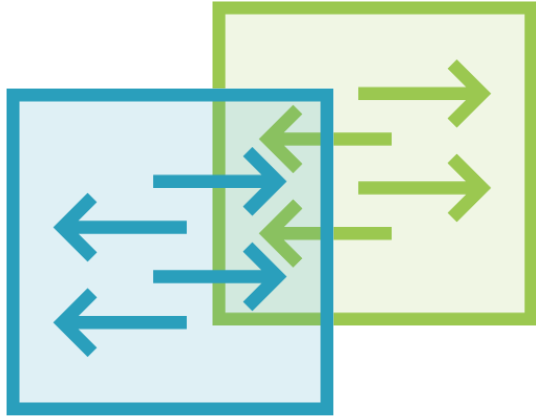
- `"api-version"=v1`

“Don’t”

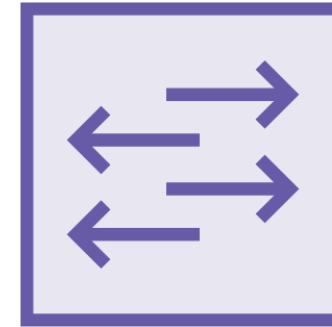
Roy Fielding (on versioning APIs, <http://bit.ly/2hBPQXi>)



# Versioning in a RESTful World



Use HATEOAS to adapt to changes  
in functionality & business rules

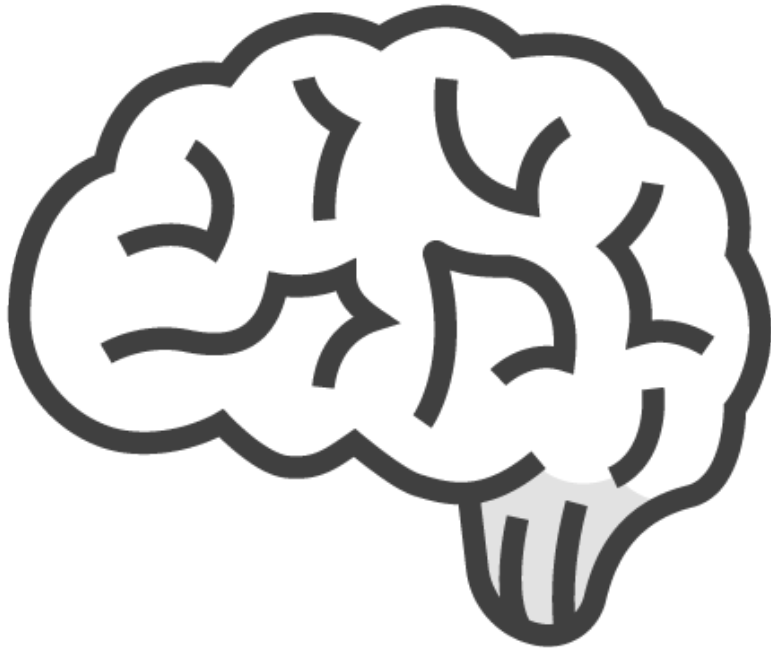


Use CoI (Code on Demand) to adapt  
to changes in media types/resource  
representations

└──┘  
Evolvability



# Versioning in a RESTful World



**Version media types to handle change in representations**

- application/vnd.marvin.author.friendly.v1+json
- application/vnd.marvin.author.friendly.v2+json

**... or use friendly names**

# Demo



## Media Types and Versioning



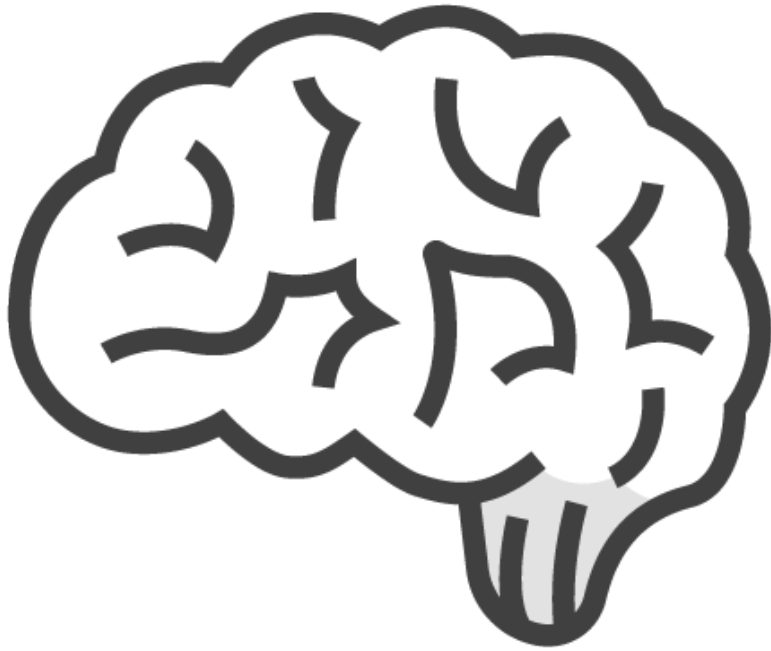
# Demo



## Supporting Multiple Media Types



# Other Approaches and Options



## HAL (Hypertext Application Language)

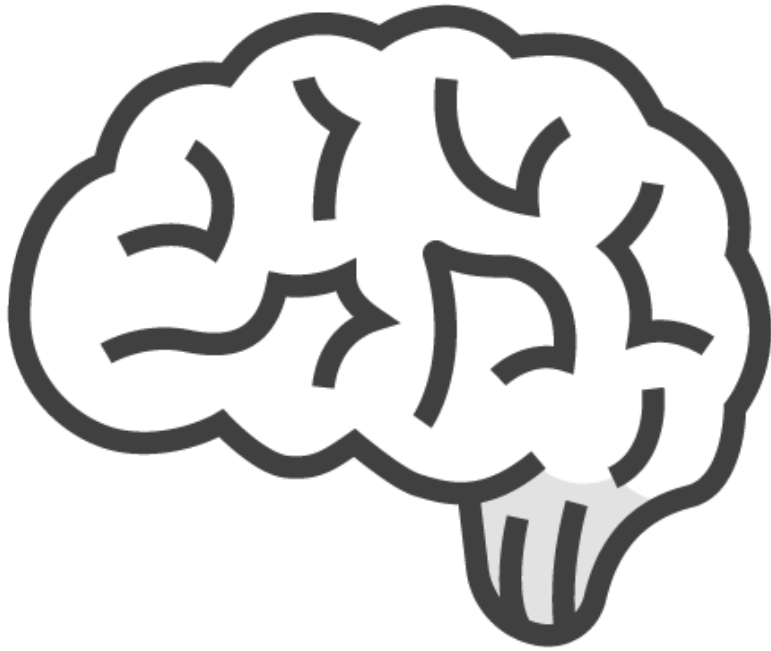
- <https://tools.ietf.org/html/draft-kelly-json-hal-08>

## SIREN (Structured Interface for Representing Entities)

- <https://github.com/kevinswiber/siren>
- <https://github.com/yury-sannikov/NHateoas> (.NET)



# Other Approaches and Options



## JSON-LD (linked data format)

- <http://json-ld.org/>

## JSON-API

- <http://jsonapi.org/>

## OData

- <http://www.odata.org/>

# Summary



## Hypermedia as the Engine of Application State

- A custom media type is warranted when the response should contain HATEOAS-related information



# Summary



## Use custom media types for resource representations

- application/vnd.marvin.author.friendly+json

## HATEOAS and media types used together lead to evolvability and solve versioning issues in a RESTful manner

- application/vnd.marvin.author.friendly.v1+json