

# An Introduction to {json:api}

Zhuo-Fei Hui  
@zfhui  
Blinkist

# Problem #1

## RESTful APIs using JSON



# Problem #1

RESTful APIs using JSON

We don't have a shared understanding  
about the structure.



## Problem #2

### Overloading

Different clients prefer different structures.

# Solution

{ json:api }

A Specification for Building APIs in JSON

{ json:api }

{ json:api }

How a client should request for resources  
to be fetched or modified.

{ json:api }

## Shared Conventions

How a **client** should request for resources  
to be fetched or modified.

How a **server** should respond to those  
requests.

# { json:api }

2013-05-03 Yehuda Katz released initial the draft

2013-07-21 registration of the media type:

application/vnd.api+json

2015-05-29 v1.0stable released

Today v1.1 still in draft

# A Simple Resource Object

# A Simple Resource Object

User(id: integer, name: string)

# Fetching a User

GET /users/1 HTTP/1.1

Accept: application/vnd.api+json

# Fetching a User

GET /users/1 HTTP/1.1

Accept: application/vnd.api+json

HTTP/1.1 200 OK

Content-Type: application/vnd.api+json,

# Fetching a User

GET /users/1 HTTP/1.1

Accept: application/vnd.api+json

HTTP/1.1 200 OK

Content-Type: application/vnd.api+json,

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" }  
  }  
}
```

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" }  
  }  
}
```

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" }  
  }  
}
```

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" }  
  }  
}
```

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" }  
  }  
}
```

# Creating a User

# Creating a User

POST /users HTTP/1.1

Accept: application/vnd.api+json

Content-Type: application/vnd.api+json

# Creating a User

POST /users HTTP/1.1

Accept: application/vnd.api+json

Content-Type: application/vnd.api+json,

```
{  
  "data": {  
    "type": "users",  
    "attributes": { "name": "Yehuda Katz" }  
  }  
}
```

# Creating a User

POST /users HTTP/1.1

Accept: application/vnd.api+json

Content-Type: application/vnd.api+json,

```
{  
  "data": {  
    "type": "users",  
    "attributes": { "name": "Yehuda Katz" }  
  }  
}
```

# Creating a User

POST /users HTTP/1.1

Accept: application/vnd.api+json

Content-Type: application/vnd.api+json,

```
{  
  "data": {  
    "type": "users",  
    "attributes": { "name": "Yehuda Katz" }  
  }  
}
```

# Creating a User

POST /users HTTP/1.1

Accept: application/vnd.api+json

Content-Type: application/vnd.api+json,

```
{  
  "data": {  
    "type": "users",  
    "attributes": { "name": "Yehuda Katz" }  
  }  
}
```

# Updating a User

PATCH /users/2

# Updating a User

PATCH /users/2,

```
{  
  "data": {  
    "id": "2",  
    "type": "users",  
    "attributes": { "name": "Dan Gebhardt" }  
  }  
}
```

# Fetching a List of Users

GET /users

GET /users

```
{  
  "data": [  
    {  
      "id": "1",  
      "type": "users",  
      "attributes": { "name": "Steve Klabnik" }  
    },  
    {  
      "id": "2",  
      "type": "users",  
      "attributes": { "name": "Dan Gebhardt" }  
    }  
  ]  
}
```

# GET /users

```
{  
  "data": [  
    {  
      "id": "1",  
      "type": "users",  
      "attributes": { "name": "Steve Klabnik" }  
    },  
    {  
      "id": "2",  
      "type": "users",  
      "attributes": { "name": "Dan Gebhardt" }  
    }  
  ]  
}
```

# GET /users

```
{  
  "data": [  
    {  
      "id": "1",  
      "type": "users",  
      "attributes": { "name": "Steve Klabnik" }  
    },  
    {  
      "id": "2",  
      "type": "users",  
      "attributes": { "name": "Dan Gebhardt" }  
    }  
  ]  
}
```

# GET /users

```
{  
  "data": [  
    {  
      "id": "1",  
      "type": "users",  
      "attributes": { "name": "Steve Klabnik" }  
    },  
    {  
      "id": "2",  
      "type": "users",  
      "attributes": { "name": "Dan Gebhardt" }  
    }  
  ]  
}
```

# Deleting a User

DELETE /users/1

# Relationships

# Relationships

User(id: integer, name: string)

# Relationships

```
User(id: integer, name: string)
```

```
Article(  
    id: integer,  
    title: string,  
    content: text,  
    user_id: integer  
)
```

# Fetching a User

GET /users/1

GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

# GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

# GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

# Compound Documents

# Compound Documents

$n+1$  requests

GET /users/1

GET /articles/2

GET /articles/5

...

# Compound Documents

n+1 requests

GET /users/1

GET /articles/2

GET /articles/5

...

1 request

# Compound Documents

n+1 requests

GET /users/1

GET /articles/2

GET /articles/5

...

1 request

GET /users/1?include=articles

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles.title

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API", "content": "Lorem opossum ..." }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding", "content": "Marsupial fur trees ..." }  
      }  
    ]  
  }  
}
```

# GET /users/1?include=articles.title

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Steve Klabnik" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "2", "type": "articles" },  
          { "id": "5", "type": "articles" }  
        ]  
      }  
    },  
    "included": [  
      {  
        "id": "2",  
        "type": "articles",  
        "attributes": { "title": "Intro to JSON API" }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding" }  
      }  
    ]  
  }  
}
```

# Updating a User with Relationships

PATCH /users/1



## PATCH /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Dan Gebhardt" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "3", "type": "articles" },  
          { "id": "6", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```



## PATCH /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Dan Gebhardt" },  
    "relationships": {  
      "articles": {  
        "data": [  
          { "id": "3", "type": "articles" },  
          { "id": "6", "type": "articles" }  
        ]  
      }  
    }  
  }  
}
```

# 👉 PATCH /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Dan Gebhardt" },  
    "relationships": {  
      "articles": {  
        "data": []  
      }  
    }  
  }  
}
```



## PATCH /users/1

```
{  
  "data": {  
    "id": "1",  
    "type": "users",  
    "attributes": { "name": "Dan Gebhardt" },  
    "relationships": {  
      "articles": {  
        "data": []  
      }  
    }  
  }  
}
```

# Manipulating a User's Relationships

# Manipulating a User's Relationships

POST and DELETE on Relationship Links:

# Manipulating a User's Relationships

POST and DELETE on Relationship Links:

/users/1/relationships/articles

# Adding Relationships to a User

# Adding Relationships to a User

POST /users/1/relationships/articles

# Adding Relationships to a User

POST /users/1/relationships/articles

```
{  
  "data": [  
    { "id": "3", "type": "articles" },  
    { "id": "7", "type": "articles" }  
  ]  
}
```

# Adding Relationships to a User

POST /users/1/relationships/articles

```
{  
  "data": [  
    { "id": "3", "type": "articles" },  
    { "id": "7", "type": "articles" }  
  ]  
}
```

# Deleting a User's Relationships

DELETE /users/1/relationships/articles

```
{  
  "data": [  
    { "id": "3", "type": "articles" },  
    { "id": "7", "type": "articles" }  
  ]  
}
```



# More Things ...

- meta objects, links objects
- pagination, sorting, filtering, sparse fieldset
- error objects
- n:m relationships
- does not support creating nested resources

# To Summarise ...

- ultimate anti-bikeshedding tool
  - one endpoint to serve different client needs
  - tight coupling between your API and the underlying data structure
-  leverages HTTP content negotiation mechanism

# Where to go from here?

gems codifying { json:api }



active\_model\_serializers



fast\_jsonapi



jsonapi\_resources



jsonapi\_suite

# Where to go from here?

Tests: [jsonapi-rspec](#)

Authorization: [jsonapi-authorization](#)

Client: [json-api-client](#) / [jsonapi-consumer](#)

# References

[Talk JSON API: convention driven API design](#) by Steve Klabnik

[Talk Past, Present and Future of JSON API](#) by Steve Klabnik

[Talk The Road to JSON API 1.0](#) by Steve Klabnik

[Website Media Type Specs](#)

[Talk "The JSON API Spec"](#) by Marco Otto-Witte

[Talk "Pragmatic JSON API Design"](#) by Jeremiah Lee

[Podcast "Dan Gebhard - json-api, jsonapi-resources, orbit.js & Ember Data"](#) by Byle Daigle

[Podcast "Data Loading Patterns with the JSON API with Balint Erdi"](#) by The Frontside Podcast

[Podcast "JSON API and API Design"](#) by The Changelog

[Image Devices](#)

A close-up photograph of a golden retriever puppy sitting on a bright green mat. The puppy is looking directly at the camera with a curious expression. In the background, there's a white ceramic dog bowl filled with kibble and a small bouquet of flowers on a shelf. The overall scene is warm and inviting.

Thanks!

# Create your new Rails API

```
rails new my-new-json-api --api
```

```
rails g scaffold user name:string
```

```
rails g scaffold article title:string content:text
```

```
rails db:setup
```

Add gem "jsonapi-resource" to your Gemfile

```
bundle install
```

# Generate JSONAPI::Resources

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user
```

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user  
create  app/resources/user_resource.rb
```

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user  
create  app/resources/user_resource.rb
```

```
class UserResource < JSONAPI::Resource; end
```

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user  
create  app/resources/user_resource.rb
```

```
class UserResource < JSONAPI::Resource
```

```
end
```

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user  
create  app/resources/user_resource.rb
```

```
class UserResource < JSONAPI::Resource  
attribute :name
```

```
end
```

# Generate JSONAPI::Resources

```
rails g jsonapi:resource user  
create  app/resources/user_resource.rb
```

```
class UserResource < JSONAPI::Resource  
  attribute :name  
  has_many :articles,  
    always_include_linkage_data: true  
end
```

```
# app/resources/article_resource.rb

class ArticleResource < JSONAPI::Resource
  attributes :title, :content
  has_one :author,
    always_include_linkage_data: true,
    foreign_key: :user_id
end
```

```
# app/resources/article_resource.rb

class ArticleResource < JSONAPI::Resource
  attributes :title, :content
  has_one :author,
    always_include_linkage_data: true,
    foreign_key: :user_id
end
```

```
# app/resources/user_resource.rb

class UserResource < JSONAPI::Resource
  attributes :name, :first_name, :last_name

  has_many :articles, always_include_linkage_data: true

  def name
    "#{@model.first_name} #{@model.last_name}"
  end

  def fetchable_fields
    %i(name articles)
  end

  def self.createable_fields(_context)
    %i(first_name last_name articles)
  end

  def self.updateable_fields(_context)
    %i(first_name last_name articles)
  end
end
```

```
# config/routes.rb
```

```
Rails.application.routes.draw do
  resources :articles
  resources :users
end
```

```
# config/routes.rb
```

```
Rails.application.routes.draw do
  resources :articles
  resources :users
end
```

```
# config/routes.rb

Rails.application.routes.draw do
  jsonapi_resources :articles
  jsonapi_resources :users
end
```

```
# app/controllers/application_controller.rb

class ApplicationController < ActionController::API

end
```

```
# app/controllers/application_controller.rb

class ApplicationController < ActionController::API
  include JSONAPI::ActsAsResourceController
end
```

```
# app/controllers/users_controller.rb

class UsersController < ApplicationController
  def index
    ...
  end

  def create
    ...
  end

  def update
    ...
  end

  ...
end
```



```
# app/controllers/users_controller.rb
```

```
class UsersController < ApplicationController; end
```

```
# app/controllers/articles_controller.rb
```

```
class ArticlesController < ApplicationController; end
```

You've just created a JSON API!



# Yehuda Katz

{ json:api } is a wire protocol for  
incrementally fetching and updating a  
graph over HTTP.

# json:api vs GraphQL

GraphQL is protocol agnostic

JSON API leverages HTTP functionalities,

such as:

# json:api vs GraphQL

GET /users/1 HTTP/1.1

Accept: application/vnd.api+json

HTTP/1.1 200 OK

Content-Type: application/vnd.api+json

ETag: "bf3291afe28105e12b9ff5941a3cf6d7"

# json:api vs GraphQL

GET /users/1 HTTP/1.1

Accept: application/vnd.api+json

If-None-Match: "bf3291afe28105e12b9ff5941a3cf6d7"

HTTP/1.1 305 Not Modified