

# An Introduction

# {json:api}

Zhuo-Fei Hui  
@zfhui  
Blinkist



# Problem

RESTful APIs using JSON

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

# Problem

## Bikeshedding

"Futile investment of time and energy in discussion of marginal technical issues."

— Wiktionary

{ json:api }

## A Specification for Building APIs in JSON

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

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

# { json:api }

"By following **shared conventions**,  
you can increase productivity, take  
advantage of generalized tooling,  
and focus on what matters: your  
application."

– jsonapi.org

# HTTP Request & Response Header

Content-Type: application/  
vnd.api+json

# A Simple Resource Object

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

# A Simple Resource Object

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

GET /users/1

# A Simple Resource Object

User(id: integer, name: string)

GET /users/1

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

# A Simple Resource Object

User(id: integer, name: string)

GET /users/1

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

# A Simple Resource Object

User(id: integer, name: string)

GET /users/1

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

# A Simple Resource Object

User(id: integer, name: string)

GET /users/1

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

## POST /users

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

# GET /users

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

# GET /users

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

# GET /users

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

# GET /users

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

PATCH /users/2

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

DELETE /users/2

# 1:n Relationship

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

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

# GET /users/1

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

# GET /articles/2

```
{  
  "data": {  
    "id": "2",  
    "type": "articles",  
    "attributes": {  
      "title": "Intro to JSON API",  
      "content": "Lorem opossum ..."  
    },  
    "relationships": {  
      "user": {  
        "data": { "id": "1", "type": "users" }  
      }  
    }  
  }  
}
```

# GET /articles/5

```
{  
  "data": {  
    "id": "5",  
    "type": "articles",  
    "attributes": {  
      "title": "Anti-Bikeshedding",  
      "content": "Marsupial fur trees ..."  
    },  
    "relationships": {  
      "user": {  
        "data": { "id": "1", "type": "users" }  
      }  
    }  
  }  
}
```

# Compound Documents

Instead of 3 requests:

GET /users/1

GET /articles/2 and GET /articles/5

We can do 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?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" }  
      },  
      {  
        "id": "5",  
        "type": "articles",  
        "attributes": { "title": "Anti-Bikeshedding" }  
      }  
    ]  
  }  
}
```

## POST /users/1

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

# POST /users/1/articles/new

```
{  
  "data": [  
    {  
      "type": "articles",  
      "attributes": { "title": "Intro to JSON API", "content": "Lorem Opossum..." }  
    },  
    {  
      "type": "articles",  
      "attributes": { "title": "Intro to JSON API", "content": "Lorem Opossum..." }  
    }  
  ]  
}
```

# PATCH /users/1

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

# More things ...

... error objects, meta objects, links  
objects, pagination, versioning, ...

# Implementation with JSONAPI::Resources

# 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
```

# gem "jsonapi-resources"

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

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

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

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

GET /users/1

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

# app/resources/user\_resource.rb

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

```
# config/routes.rb

Rails.application.routes.draw do
  # resources :articles
  # resources :users

  jsonapi_resources :articles
  jsonapi_resources :users
end
```

# Versioning

JSON API is strictly additive

# Authorization

# Testing

# History

2013-05-03 Yehuda Katz released initial draft

2013-07-21 media type registration with  
IANA completed

2014-07-05 v1.0rc released

2015-05-29 v1.0stable released

Today v1.1 still in draft

# References

Talk "The JSON API Spec" by  
Marco Otto-Witte

# 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