

PONAPI

Mickey Nasriachi

Oct-2015

JSON-API

JSONAPI.ORG

JSONAPI

JSONAPI

PONAPI

Bikeshedding

~~Bikeshedding~~

the spec

queries

GET /type/id?query_parameters

GET /articles

```
{
  "jsonapi" : {
    "version" : "1.0"
  },
  "data" : [
    {
      "attributes" : {
        "title" : "a third one",
        "status" : "pending approval",
        "created" : "2015-07-22T14:56:29.000Z",
        "updated" : "2015-07-22T14:56:28.000Z",
        "body" : "The 3rd shortest article. Ever."
      },
      "relationships" : {
        "author" : {
          "data" : {
            "type" : "people",
            "id" : "91"
          }
        }
      },
      "type" : "articles",
      "id" : "3"
    },
  ],
}
```



```
{  
  "relationships" : {
```

GET /articles/3

```
{
  "jsonapi" : {
    "version" : "1.0"
  },
  "data" : {
    "id" : "3",
    "attributes" : {
      "created" : "2015-07-22T14:56:29.000Z",
      "body" : "The 3rd shortest article. Ever.",
      "status" : "pending approval",
      "updated" : "2015-07-22T14:56:28.000Z",
      "title" : "a third one"
    },
    "relationships" : {
      "author" : {
        "data" : {
          "id" : "91",
          "type" : "people"
        }
      }
    }
  },
  "type" : "articles"
}
```

}

GET /articles/3?include=author

```
{
  "jsonapi" : {
    "version" : "1.0"
  },
  "included" : [
    {
      "attributes" : {
        "age" : "30",
        "name" : "Diana",
        "gender" : "female"
      },
      "type" : "people",
      "id" : "91"
    }
  ],
  "data" : {
    "id" : "3",
    "type" : "articles",
    "relationships" : {
      "author" : {
        "data" : {
          "id" : "91",
          "type" : "people"
        }
      }
    }
  }
}
```

GET /articles/3?fields[articles]=body,title

```
{
  "jsonapi" : {
    "version" : "1.0"
  },
  "data" : {
    "attributes" : {
      "body" : "The 3rd shortest article. Ever.",
      "title" : "a third one"
    },
    "id" : "3",
    "type" : "articles"
  }
}
```


GET /articles/99/author

```
{
  "data" : {
    "id" : "91",
    "type" : "people",
    "attributes" : {
      "gender" : "female",
      "age" : "30",
      "name" : "Diana"
    }
  },
  "jsonapi" : {
    "version" : "1.0"
  }
}
```

implementation

Moose

```
with 'PONAPI::Builder',
     'PONAPI::Builder::Role::HasLinksBuilder',
     'PONAPI::Builder::Role::HasMeta';

has '_included' => (
  init_arg => undef,
  traits   => [ 'Array' ],
  is       => 'ro',
  isa      => 'ArrayRef[ PONAPI::Builder::Resource ]',
  lazy     => 1,
  default  => sub { +[] },
  handles  => {
    'has_included' => 'count',
    # private ...
    '_add_included' => 'push',
  }
);

...

__PACKAGE__->meta->make_immutable;
no Moose; 1;
```

Server (Dancer2)

```
use Dancer2;

use Dancer2::Plugin::JSONAPI::MediaType;
use Dancer2::Plugin::JSONAPI::Params;
use Dancer2::Plugin::PONAPI::Repository;

set serializer => 'JSON';

# no ID
prefix '/:resource_type' => sub {

    # Create new resource(s)
    post '' => sub { DAO->create( jsonapi_parameters ) };

    # Retrieve all resources for type
    get '' => sub { DAO->retrieve_all( jsonapi_parameters ) };
};

# required ID
prefix '/:resource_type/:resource_id' => sub {

    # Retrieve a single resource
    get '' => sub { DAO->retrieve( jsonapi_parameters ) };

    # Retrieve related resources indirectly
```

DAO

Client

```
use strict;
use warnings;

use PONAPI::Client;

my $cl = PONAPI::Client->new();

my $res = $cl->retrieve(
    type    => 'articles',
    id      => 3,
    fields => { articles => [qw< body title author >] }
);
```


