

{ json:api }

A SPECIFICATION FOR BUILDING APIS IN JSON

View the specification
<<http://jsonapi.org/format/>>

Contribute on GitHub
<<https://github.com/json-api/json-api>>

If you've ever argued with your team about the way your JSON responses should be formatted, JSON API can be your anti-[bikeshedding](http://bikeshed.org) <<http://bikeshed.org>> tool.

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

Clients built around JSON API are able to take advantage of its features around efficiently caching responses, sometimes eliminating network requests entirely.

Here's an example response from a blog that implements JSON API:

```
{
  "links": {
    "self": "http://example.com/articles",
    "next": "http://example.com/articles?page[offset]=2",
    "last": "http://example.com/articles?page[offset]=10"
  },
  "data": [{
    "type": "articles",
    "id": "1",
    "attributes": {
      "title": "JSON API paints my bikeshed!"
    },
    "relationships": {
      "author": {
        "links": {
          "self": "http://example.com/articles/1/relationships/author",
```

```

    "related": "http://example.com/articles/1/author"
  },
  "data": { "type": "people", "id": "9" }
},
"comments": {
  "links": {
    "self": "http://example.com/articles/1/relationships/comments",
    "related": "http://example.com/articles/1/comments"
  },
  "data": [
    { "type": "comments", "id": "5" },
    { "type": "comments", "id": "12" }
  ]
},
"links": {
  "self": "http://example.com/articles/1"
}
}],
"included": [{
  "type": "people",
  "id": "9",
  "attributes": {
    "first-name": "Dan",
    "last-name": "Gebhardt",
    "twitter": "dgeb"
  },
  "links": {
    "self": "http://example.com/people/9"
  }
}, {
  "type": "comments",
  "id": "5",
  "attributes": {
    "body": "First!"
  },
  "relationships": {
    "author": {
      "data": { "type": "people", "id": "2" }
    }
  },
  "links": {
    "self": "http://example.com/comments/5"
  }
}, {
  "type": "comments",
  "id": "12",
  "attributes": {
    "body": "I like XML better"
  },
  "relationships": {
    "author": {
      "data": { "type": "people", "id": "9" }
    }
  },
  "links": {
    "self": "http://example.com/comments/12"
  }
}

```

```
}  
  }]  
}
```

The response above contains the first in a collection of “articles”, as well as links to subsequent members in that collection. It also contains resources linked to the article, including its author and comments. Last but not least, links are provided that can be used to fetch or update any of these resources.

JSON API covers creating and updating resources as well, not just responses.

MIME Types

JSON API has been properly registered with the IANA. Its media type designation is [application/vnd.api+json](http://www.iana.org/assignments/media-types/application/vnd.api+json) <<http://www.iana.org/assignments/media-types/application/vnd.api+json>>.

Format documentation

To get started with JSON API, check out [documentation for the base specification](http://jsonapi.org/format) <<http://jsonapi.org/format>>.

Extensions

JSON API has [experimental support for extensions](http://jsonapi.org/extensions) <<http://jsonapi.org/extensions>>.

Official extensions are being developed for [Bulk](http://jsonapi.org/extensions/bulk/) <<http://jsonapi.org/extensions/bulk/>> and [JSON Patch](http://jsonapi.org/extensions/jsonpatch/) <<http://jsonapi.org/extensions/jsonpatch/>> operations.

Update history

- 2015-05-29: 1.0 final released.
- 2015-05-21: Release candidate 4 released.
- 2015-03-16: Release candidate 3 released.
- 2015-02-18: Release candidate 2 released.
- 2014-07-05: Release candidate 1 released.
- 2013-07-21: Media type registration completed with the IANA.
- 2013-05-03: Initial release of the draft.

You can subscribe to an RSS feed of individual changes [here <https://github.com/json-api/json-api/commits.atom>](https://github.com/json-api/json-api/commits.atom).