# How are REST APIs versioned?

Posted on March 12, 2012 by Tim Wood, Modified on September 15, 2014

I am currently working on a REST API, and the question was raised, how are, and how should, REST APIs be versioned? Here are the results of my research.

It seems that there are a number of people recommending using Content-Negotiation (the HTTP "Accept:" header) for API versioning. However, none of the big public REST APIs I have looked at seem to be using this approach. They almost exclusively put the API version number in the URI, with the odd exception using a custom HTTP header. I am at somewhat of a loss to explain this disconnect.

### Versioning strategies in discussions

| POST | VERSIONING |
|---|---|
| Stack Overflow 1 | URI |
| Stack Overflow 2 | Content Negotiation |
| blog post by Jeremy | Content Negotiation |
| ycombinator discussion | some opinions both ways |
| Stack Overflow 3 | Content Negotiation |
| Stack Overflow 4 | Content Negotiation |
| notmessenger blog post | URI (against all headers) |
| Peter Williams | Content Negotiation (strongly against URIs) |
| Apigee Blog post on API versioning | URI (some discussion) |
| Mark Nottingham REST versioning | Recommending essentially versionless extensibility with a HATEOS approach |
| Nick Berardi on REST versioning | URI |
| Restify | "Accept-Version" header |
| Tom Maguire on REST versioning | Content Negotiation |
| Nicholas Zakas on Rest Versioning | URI |
| Steve Klabnik on Rest Versioning | Content Negotiation + HATEOS |
| Luis Rei on Rest Versioning | Content Negotiation with (;version=1.0) |
| kohana forum discussion on REST versioning | Many Opinions |
| Troy Hunt on REST versioning | Accept Header but also support custom header and URL |
| Paul Gear REST versioning | Recommending essentially versionless extensibility with a HATEOS approach |

### Versioning strategies in popular REST APIs

| API NAME | VERSIONING | EXAMPLE |
|---|---|---|
| Twillo | date in URI | |
| Twitter | URI | |
| Atlassian | URI | |
| Google Search | URI | |
| Github API | URI/Media Type in v3 | Intention is to remove versioning in favour of hypermedia – current application/vnd.github.v3 |
| Azure | Custom Header | x-ms-version: 2011-08-18 |
| Facebook | URI/optional versioning | graph.facebook.com/v1.0/me |
| Bing Maps | URI | |
| Google maps | unknown/strange | |
| Netflix | URI parameter | http://api.netflix.com/catalog/titles/series/70023522?v=1.5 |
| Salesforce | URI with version introspection | {<br>“label”:”Winter ’10”<br>“version”:”20.0″,<br>“url”:”/services/data/v20.0″,<br>} |
| Google data API (youtube/spreadsheets/others) | URI parameter or custom header | “GData-Version: X.0” or “v=X.0” |
| Flickr | No versioning? | |
| Digg | URI | http://services.digg.com/2.0/comment.bury |
| Delicious | URI | https://api.del.icio.us/v1/posts/update |
| Last FM | URI | http://ws.audioscrobbler.com/2.0/ |
| LinkedIn | URI | http://api.linkedin.com/v1/people/~/connections |
| Foursquare | URI | https://api.foursquare.com/v2/venues/40a55d80f964a52020f31ee3?oauth_token=XXX&v=YYYYMMDD |
| Freebase | URI | https://www.googleapis.com/freebase/v1/search?query=nirvana&indent=true |
| paypal | parameter | &VERSION=XX.0 |
| Twitpic | URI | http://api.twitpic.com/2/upload.format |
| Etsy | URI | http://openapi.etsy.com/v2 |
| Tropo | URI | https://api.tropo.com/1.0/sessions |
| Tumblr | URI | api.tumblr.com/v2/user/ |
| openstreetmap | URI and response body | http://server/api/0.6/changeset/create |

| | | |
|---|---|---|
| Ebay | URI (I think) | http://open.api.ebay.com/shopping?version=713 |
| Reddit | No versioning? | |
| Groupon | URI | http://api.groupon.com/v2/channels//deals{.json\|.xml} |
| Geonames | | |
| Wikipedia | no versioning I think? | |
| Bitly | URI | https://api-ssl.bitly.com/v3/shorten |
| Disqus | URI | https://disqus.com/api/3.0/posts/remove.json |
| Yammer | URI | /api/v1 |
| Drop Box | URI | https://api.dropbox.com/1/oauth/request_token |
| Amazon Simple Queue Service (Soap) | URI Parameter and WSDL URI | &Version=2011-10-01 |
| Youtube data API versioning | URI | https://www.googleapis.com/youtube/v3 |

Versioning strategies in popular REST Libraries

| LIBRARY NAME | VERSIONING | EXAMPLE |
|---|---|---|
| node-restify | semver versioning in an accept-Version header | accept-version: ~3 |
| Jersey | description of how to do Accept: header versioning in Jersey | Accept: application/vnd.musicstore-v1+json |

No related posts.