Can comments be used in JSON?

Asked 10 years, 11 months ago Active 10 days ago Viewed 1.9m times

6952

711

Can I use comments inside a JSON file? If so, how?

json comments

edited Jul 22 '16 at 15:39
Steve Chambers
23.5k 12 108 152

asked Oct 28 '08 at 20:39
Michael Gundlach
40.4k 11 33 38

133 @StingyJack: To explain things that may not be obvious, or whatever else one might do with comments. I for one often have comments in data files. XML, ini files, and many other formats include provisions for comments. – Michael Burr Oct 28 '08 at 20:51

21 If you, like me, were wondering whether `//comments` are OK for the specific use-case of a Sublime Text configuration file, the answer is yes (as of version 2). Sublime Text will not complain about it, at least, whereas it will complain about `{"__comment": ...}` in the console, because it is an unexpected field. – driftcatcher Feb 1 '13 at 15:12

6 and perhaps this is one reason why TOML was created.. – Alex Nolasco May 1 '13 at 5:22

8 Slightly noobish but ,i also tried using // for comments in JSON. Now I realize it is strictly used for interchange/exchange. Sigh! I cant comment any more :(. Life is doomed!. – Sid Sep 25 '13 at 11:29

9 JSON5 supports comments: stackoverflow.com/a/7901053/108238 – schoetbi Feb 2 '15 at 11:13

49 Answers

1 2 next

No.

The JSON should all be data, and if you include a comment, then it will be data too.

You would probably be better having the comment in the processes that generates/receives the JSON, as they are supposed to know what the JSON data will be in advance, or at least the structure of it.

But if you decided to:

```
{
   "_comment": "comment text goes here...",
   "glossary": {
      "title": "example glossary",
      "GlossDiv": {
         "title": "S",
         "GlossList": {
            "GlossEntry": {
               "ID": "SGML",
               "SortAs": "SGML",
               "GlossTerm": "Standard Generalized Markup Language",
               "Acronym": "SGML",
               "Abbrev": "ISO 8879:1986",
               "GlossDef": {
                  "para": "A meta-markup language, used to create markup languages such
as DocBook.",
                  "GlossSeeAlso": ["GML", "XML"]
               },
               "GlossSee": "markup"
            }
         }
      }
   }
}
```

edited Apr 3 '18 at 11:57
Christian Neverdal
2,849 2 26 82

answered Oct 28 '08 at 21:01
Eli
65k 19 69 81

204 It might pay to have some kind of prefix on the actual comment in case there's ever a valid field named comment: `"__comment":"comment text goes here...",` – Rob Fonseca-Ensor Feb 3 '10 at 11:41

17 BTW, the json library for Java google-gson has support for comments. – centic Oct 1 '12 at 12:21

9 What about if I wanted a separate comment on the `Accronym` and `Abbrev` properties? I've used this pattern before but stopped since it doesn't allow me to do that. It is a hack. Maybe if I prepend a property name with `__comment__` instead. That is "__comment__Abbrev", still a hack, but would let me comment on all prpoerties – Juan Mendes Aug 11 '14 at 12:58

2 One of the links in my June 2012 comment no longer works. Another reader (@douglasgross) has provided this current link: groups.yahoo.com/neo/groups/json/conversations/topics/156 – Michael Burr Sep 27 '18 at 22:20

1745

No, comments of the form `//…` or `/*…*/` are not allowed in JSON. This answer is based on:

- http://www.json.org
- RFC 4627: The `application/json` Media Type for JavaScript Object Notation (JSON)
- RFC 8259 The JavaScript Object Notation (JSON) Data Interchange Format (supercedes RFCs 4627, 7158, 7159)

edited Jul 10 at 17:22

answered Nov 15 '10 at 9:32
stakx
66.9k 16 139 229

59 If you'd like to annotate your JSON with comments (thus making it invalid JSON), then minify it before parsing or transmitting. Crockford himself acknowledged this in 2012 in the context of configuration files. – toolbear Aug 7 '14 at 19:26

23 @alkuzad: When it comes to formal grammars, there must be something that explicitly says that they *are* allowed, not the other way around. For instance, take your programming language of choice: Just because some desired (but missing) feature isn't explicitly disallowed, doesn't mean that your compiler will magically recognize it. – stakx Sep 28 '15 at 7:01

3 Yes. The JSON format has a lot of dead-space between elements and is space-insensitive in those regions, so there's no reason why you can't have single or multi-line comments there. Many parsers and minifiers support JSON comments as well, so just make sure your parser supports them. JSON is used a lot for application data and configuration settings, so comments are necessary now. The "official spec" is a nice idea, but it's insufficient and obsolete, so too bad. Minify your JSON if you're concerned about payload size or performance. – Triynko Oct 31 '17 at 17:36

43 Although your answer is absolutely correct, it should be said that this is BS. With so many end users coming across the need for json configuration, then comments are exceedingly helpful. Just because some tin-foil hats decided that JSON *is* and *must always be* machine readable, ignoring the fact that humans needs to read it to, is imho a travesty of small mindedness. – cmroanirgo Jan 21 '18 at 0:29

11 @cmroanirgo: You're obviously not the first to complain about that limitation of JSON... that's why we have parsers that silently allow comments, and other formats such as YAML and JSON5. However this doesn't change the fact that JSON is what it is. Rather, I find it interesting that people started using JSON for purposes where it clearly wasn't sufficient in the first place, given the limitation in question. Don't blame the JSON format; blame ourselves for insisting on using it where it isn't a particularly good fit. – stakx Jan 21 '18 at 7:48

Include comments if you choose; strip them out with a minifier before parsing or transmitting.

```
JSON.parse(JSON.minify(my_str));
```

When I released it, I got a huge backlash of people disagreeing with even the idea of it, so I decided that I'd write a comprehensive blog post on why comments make sense in JSON. It includes this notable comment from the creator of JSON:

> Suppose you are using JSON to keep configuration files, which you would like to annotate. Go ahead and insert all the comments you like. Then pipe it through JSMin before handing it to your JSON parser. - Douglas Crockford, 2012

Hopefully that's helpful to those who disagree with why **JSON.minify()** could be useful.

edited Jan 20 '17 at 13:40

answered Jun 23 '10 at 18:20
Kyle Simpson
12.3k 2 26 45

145 The only problem I have with JSON.minify() is that it is really really slow. So I made my own implementation that does the same thing: gist.github.com/1170297 . On some large test files your implementation takes 74 seconds and mine 0.06 seconds. – WizKid Aug 25 '11 at 9:16

53 it'd be great if you could submit the suggested alternative algorithm to the github repo for JSON.minify(), so that it can be ported to all the supported langs: github.com/getify/json.minify – Kyle Simpson Aug 30 '11 at 17:20

16 @MiniGod I have already heard Doug's thoughts on this topic many times. I addressed them long ago in my blog post: blog.getify.com/json-comments – Kyle Simpson Feb 26 '13 at 23:21

17 @MarnenLaibow-Koser there are still valid uses for comments even for data stream (or even packet) usage: inclusion of diagnostics metadata like creation time or sources is common use with XML, and perfectly sensible for JSON data as well. Arguments against comments are shallow, and any textual data format should allow for comments, regardless of implied intended usage (nothing spec suggest JSON can not be used elsewhere, fwiw) – StaxMan Mar 28 '14 at 22:09

15 If JSON is to have universal acceptance (which it basically does) then it should have universal application. Example: JSON can serve as an application configuration file. This application would desire comments. – eggmatters Jan 27 '16 at 19:31

416

Comments were removed from JSON by design.

> I removed comments from JSON because I saw people were using them to hold parsing directives, a practice which would have destroyed interoperability. I know that the lack of comments makes some people sad, but it shouldn't.

Source: Public statement by Douglas Crockford on G+

edited Jul 30 '12 at 21:48
hakre
164k 33 324 643

answered Jun 11 '12 at 8:52
Artur Czajka
12.4k 2 21 28

174 I thought JSON was to supposed to be more human readable than, say, XML? Comments are for readability. – intrepidis Oct 14 '12 at 13:00

23 Anyway, you could be naughty and add parsing directives in the JSON: {"__directives":{"#n#":"DateTime.Now"}, "validdate":"#n#"}... It looks like YAML is the way forward then... – intrepidis Oct 14 '12 at 13:04

64 Personal opinion: not allowing comments IS lame. I had no option other than building a non-standard JSON parser that ignores comments, to decode my config files. – caiosm1005 Sep 23 '13 at 2:28

17 @ArturCzajka I still dislike the fact JSON doesn't support comments, but I gave INI a try and I must admit it makes much more sense to use them over JSON for config files. Thanks for the response and hopefully more people will change their minds as they read this conversation. (making a parser was more of an exercise anyway :) – caiosm1005 Oct 4 '13 at 2:42

66 That's like requiring all bicycles to have training wheels because some people can't ride bicycles. Removing an important feature because stupid people abuse it is bad design. A data format should prioritize usability over being idiot-proof. – Phil Goetz May 14 '15 at 17:24

199

DISCLAIMER: YOUR WARRANTY IS VOID

As has been pointed out, this hack takes advantage of the implementation of the spec. Not all JSON parsers will understand this sort of JSON. Streaming parsers in particular will choke.

It's an interesting curiosity, but you **should really not be using it for anything at all**. Below is the original answer.

I've found a little hack that allows you to place comments in a JSON file that will not affect the parsing, or alter the data being represented in any way.

It appears that when declaring an object literal you can specify two values with the same key, and the last one takes precedence. Believe it or not, it turns out that JSON parsers work the same way. So we can use this to create comments in the source JSON that will not be present in a parsed object representation.

```
Object.keys(JSON.parse('{"a": 1, "a": 2}')).length;
// => 1
```

If we apply this technique, your commented JSON file might look like this:

```
{
  "api_host" : "The hostname of your API server. You may also specify the port.",
  "api_host" : "hodorhodor.com",

  "retry_interval" : "The interval in seconds between retrying failed API calls",
  "retry_interval" : 10,

  "auth_token" : "The authentication token. It is available in your developer dashboard
under 'Settings'",
  "auth_token" : "5ad0eb93697215bc0d48a7b69aa6fb8b",

  "favorite_numbers": "An array containing my all-time favorite numbers",
  "favorite_numbers": [19, 13, 53]
}
```

The above code is valid JSON. If you parse it, you'll get an object like this:

```
{
    "api_host": "hodorhodor.com",
    "retry_interval": 10,
    "auth_token": "5ad0eb93697215bc0d48a7b69aa6fb8b",
    "favorite_numbers": [19,13,53]
}
```

Which means there is no trace of the comments, and they won't have weird side-effects.

Happy hacking!

edited Jan 15 '16 at 5:58
thanksd
29.9k 15 86 93

answered Aug 2 '13 at 13:46
p3drosola
4,874 2 18 29

146 From the specification: The names within an object SHOULD be unique. – Quentin Aug 2 '13 at 13:50

104 "all the implementations handle it the same" – That's a difficult thing to prove. – Quentin Aug 2 '13 at 14:30

64 This clearly violates the spec (see above comments), don't do this. ietf.org/rfc/rfc4627.txt?number=4627 – voidlogic Aug 2 '13 at 14:39

322 **NO** - what if the parser is streaming? What if the parser reads it into a dictionary where key ordering is undefined? **kill this with fire**. – deanWombourne Aug 2 '13 at 14:55

167

JSON does not support comments. It was also never intended to be used for configuration files where comments would be needed.

Hjson is a configuration file format for humans. Relaxed syntax, fewer mistakes, more comments.

See hjson.org for JavaScript, Java, Python, PHP, Rust, Go, Ruby and C# libraries.

edited Aug 9 '16 at 21:20

answered Mar 20 '14 at 15:26
laktak
28.5k 12 100 141

12 Upvoted. It's obviously a good variation un-open conservative people would just love to hate. I hope your implementation gets known further - and perhaps even gets more popular than the original ;) I hope someone gets to implement it with Ruby as well. @adelphus The language being well-defined is your own perspective or opinion. Being a conservative "developer" if you are one doesn't prove that you are better and you could be even worse keeping yourself locked up in limited spaces. Don't go judging people as terrible developers easily. – konsolebox Jun 30 '14 at 19:20

7 Sorry about that, @konsolebox. Perhaps you might reconsider your "well-defined JSON is your opinion" view after reading ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf It is a real standard and devs implementing their own "special" versions leads to

implements slightly different versions of standards. The JSON language may not be perfect, but fragmentation is worse. And yes, that's just a opinion and you're free to disagree. – adelphus Jul 9 '14 at 16:02

20 I admire your gumption, but you're kinda re-inventing YAML. If you want lot's of flexibility and human readability, use YAML (don't actually: stackoverflow.com/questions/450399/…) or stick with curmudgeony, yet unambiguous JSON. – toolbear Aug 7 '14 at 18:25

4 I find the most user-friendly configuration format is still INI. It's straightforward and not very syntax heavy. This makes it less intimidating for users just dipping their toes in the configuration pond. – Matt Feb 10 '15 at 15:15

13 Whenever you need json as config (where comments *are* needed) - name your file ".js" instead of ".json".. js can of course handle any valid json object and additionally *can* handle comments.. That's the reason why it is "webpack.config.js" and not "webpack.config.json" (well there's a lot more reasons for that too in webpack :P) – jebbie Apr 6 '16 at 14:20

113

Consider using YAML. It's nearly a superset of JSON (virtually all valid JSON is valid YAML) and it allows comments.

answered Aug 31 '11 at 2:24
Marnen Laibow-Koser
4,422 1 18 30

12 @g33kz0r Correct, hence my description of YAML as a near-superset of JSON. – Marnen Laibow-Koser Sep 12 '12 at 14:58

12 @NateS Many people had already pointed out that the answer was no. I suggested a better way to achieve the OP's goal. That's an answer. – Marnen Laibow-Koser Mar 28 '14 at 12:57

4 Downside: `yaml` library isn't shipped with Python. – Bleeding Fingers Apr 15 '14 at 10:06

7 @BleedingFingers So install it... – Marnen Laibow-Koser Apr 15 '14 at 16:19

5 @marnen-laibow-koser: yup, it must have been incompetence to use the available YAML libraries for Java and Perl and expect the YAML produced by each to be consumed by the other without error. That YAML interop was an issue, but JSON interop wasn't, is entirely explained by my lack of knowledge. – toolbear Aug 17 '14 at 23:32

102

You can't. At least that's my experience from a quick glance at json.org.

JSON has its syntax visualized on that page. There isn't any note about comments.

edited Jan 19 '17 at 17:20 answered Oct 28 '08 at 20:42

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

66

You should write a JSON schema instead. JSON schema is currently a proposed Internet draft specification. Besides documentation, the schema can also be used for validating your JSON data.

Example:

```
{
    "description":"A person",
    "type":"object",
    "properties":
        {
            "name":
                {
                    "type":"string"
                },
            "age":
                {
                    "type":"integer",
                    "maximum":125
                }
        }
}
```

You can provide documentation by using the **description** schema attribute.

edited Jan 19 '17 at 17:37
Peter Mortensen
14.5k 19 89 118

answered Jul 28 '10 at 18:38
raffel
1,071 9 8

5 Is JSON schema alive? It exists but is it supported by any known library? – Munhitsu Oct 29 '12 at 14:31

1 yes, the json-schema google group is fairly active and I would recommend JSV for a good JavaScript implementation of a JSON Schema validator. – raffel Nov 27 '12 at 11:34

5 This only helps with structured documentation, not ad-hoc documentation – Juan Mendes Apr 4 '13 at 17:47

If you use clojure (and I'm sure you don't) there's a reasonably featured open-source JSON schema parser here: github.com/bigmlcom/closchema – charlesIparker Apr 15 '13 at 13:50

@Munhitsu Manatee.Json (.Net) extensively supports JSON schema. – gregsdennis Jun 26 '15 at 0:26

60

If you are using Jackson as your JSON parser then this is how you enable it to allow comments:

```
ObjectMapper mapper = new ObjectMapper().configure(Feature.ALLOW_COMMENTS, true);
```

Then you can have comments like this:

```
{
  key: "value" // Comment
}
```

And you can also have comments starting with `#` by setting:

```
mapper.configure(Feature.ALLOW_YAML_COMMENTS, true);
```

But in general (as answered before) the specification does not allow comments.

edited Oct 14 '18 at 17:03
Peter Mortensen
14.5k 19 89 118

answered Feb 6 '14 at 20:44
Andrejs
20.9k 9 91 87

58

Comments are not an official standard. Although some parsers support C-style comments. One that I use is JsonCpp. In the examples there is this one:

```
// Configuration options
{
    // Default encoding for text
    "encoding" : "UTF-8",

    // Plug-ins loaded at start-up
    "plug-ins" : [
        "python",
        "c++",
        "ruby"
        ],
```

jsonlint does not validate this. So comments are a parser specific extension and not standard.

Another parser is JSON5.

An alternative to JSON TOML.

A further alternative is jsonc.

edited Apr 3 at 8:12
Hotschke
5,576 4 29 41

answered Oct 26 '11 at 9:46
schoetbi
6,207 5 39 66

Groovy has some built-in classes for handling JSON. JsonSlurper can handle comments. Of course, comments are not allowed in the official spec, so this behavior in any parser is non-standard and non-portable. – izrik Nov 6 '15 at 21:47

48

Here is what I found in the Google Firebase documentation that allows you to put comments in JSON:

```
{
  "//": "Some browsers will use this to enable push notifications.",
  "//": "It is the same for all projects, this is not your project's sender ID",
  "gcm_sender_id": "1234567890"
}
```

edited Jun 22 '17 at 13:05

answered Jun 22 '17 at 12:58
mana
4,140 3 36 63

FYI, Firebase Realtime Database does not allow the use of '/' in a key. so this can be a nice convention for your own use, but you cannot do it in Firebase – gutte Nov 1 '17 at 20:44

3 This method breaks some libraries, which require that the key must be unique. I'm working around that issue by numbering the comments. – MovGP0 Jan 19 '18 at 11:45

good comment, I found this question on SO ... this part seems not to be covered by the spec stackoverflow.com/questions/21832701/… – mana Jan 20 '18 at 19:36

3 I tend to use it like this nowadays: { "//foo": "foo comment", "foo": "foo value", "//bar": "bar comment", "bar": "bar value" } You can use an array for multiple comments: { "//foo": ["foo comment 1", "foo comment 2"], "foo": "foo value" } – MovGP0 Mar 9 '18 at 16:29

38

If your text file, which is a JSON string, is going to be read by some program, how difficult would it be to strip out either C or C++ style comments before using it?

Answer: It would be a one liner. If you do that then JSON files could be used as configuration files.

edited Aug 29 '15 at 17:51
Randika Vishman
5,452 3 45 62

answered Apr 9 '10 at 22:30
John T. Vonachen
445 4 2

1 Probably the best suggestion so far, though still an issue for keeping files as an interchange format, as they need pre-processing before use. – Orbling Feb 25 '11 at 11:04

I agree and have written a JSON parser in Java, available at www.SoftwareMonkey.org, that does exactly that. – Lawrence Dol Jul 28 '12 at 1:51

2 Despite I think, it is not a good idea to extend JSON (without calling it a different exchange format): make sure to ignore "comments" within strings. { "foo": "/* This is not a comment.*/" } – stofl Jul 27 '13 at 12:09

24 "...would be a one liner" umm, no, actually, JSON is not a regular grammar where a regular expression can simply find matching pairs of /*. You have to parse the file to find if a /* appears inside a string (and ignore it), or if it's escaped (and ignore it), etc. Also, your answer is unhelpful because you simply speculate (incorrectly) rather than providing any solution. – Kyle Simpson Dec 8 '13 at 21:55

1 What @kyle-simpson said. Also, he's too modest to direct readers to his own answer about using JSON.minify as an alternative to ad hoc regexps. Do that, not this. – toolbear Aug 7 '14 at 19:32

34

If you are using the Newtonsoft.Json library with ASP.NET to read/deserialize you can use comments in the JSON content:

> //"name": "string"
>
> //"id": int

or

> comment example */

PS: Single-line comments are only supported with 6+ versions of Newtonsoft Json.

Additional note for people who can't think out of the box: I use the JSON format for basic settings in an ASP.NET web application I made. I read the file, convert it into the settings object with the Newtonsoft library and use it when necessary.

I prefer writing comments about each individual setting in the JSON file itself, and I really don't care about the integrity of the JSON format as long as the library I use is OK with it.

I think this is an 'easier to use/understand' way than creating a separate 'settings.README' file and explaining the settings in it.

If you have a problem with this kind of usage; sorry, the genie is out of the lamp. People would find other usages for JSON format, and there is nothing you can do about it.

edited Jan 19 '17 at 18:21
Peter Mortensen
14.5k 19 89 118

answered Jul 25 '14 at 13:43
dvdmn
4,047 5 37 49

It is hard to understand why someone would have problem with stating a fact. – dvdmn Jul 29 '14 at 14:17

I would assume someone took exception because the above is no longer JSON, or is invalid JSON. Perhaps adding a short disclaimer would appease. – toolbear Aug 7 '14 at 18:12

5 I completely agree with you, and yet there are 883 upvotes so far for the non-answer that just states the obvious. Ideological purity valued above helpful information, that's SO for you. – John Aug 20 '14 at 16:48

The point is a file with comments is not JSON and will fail to be parsed by many JSON libraries. Feel free to do whatever you want in your own program but a file with comments is not JSON. If you claim it is then people will try to parse it with their language/library of choice and it will fail. It's like asking if you can use square brackets instead of angle brackets in XML. You can do whatever you want but it will no longer be XML. – gman Jun 6 at 3:35

30

The idea behind JSON is to provide simple data exchange between applications. These are typically web based and the language is JavaScript.

It doesn't really allow for comments as such, however, passing a comment as one of the name/value pairs in the data would certainly work, although that data would obviously need to be ignored or handled specifically by the parsing code.

edited Jan 19 '17 at 17:25
Peter Mortensen
14.5k 19 89 118

answered Oct 28 '08 at 23:05
Neil Albrock
705 8 16

14 It is true that JSON format does not have comments. Personally I think that is a significant mistake -- ability to have comments as metadata (not data) is a very useful thing with xml. Earlier draft versions of JSON specification did include comments, but for some reason they were dropped. :-/ – StaxMan Sep 1 '09 at 18:20

4 @StaxMan they were dropped exactly because people started using them as metadata. Crockford said it breaked the compatibility for what the format was designed, and I agree: if you want metadata, why not include it as actual data? It's even easier to parse this way. – Camilo Martin Dec 11 '10 at 9:03

6 Metadata belongs in metadata constructs (e.g. HTML <meta> tags), not comments. Abusing comments for metadata is just a hack used where no true metadata construct exists. – Marnen Laibow-Koser Sep 6 '11 at 4:55

That's exactly the reason why it was dropped: comments used as metadata would break interoperability. You should just store your meta-data as JSON too. – gaborous Jun 25 '13 at 14:50

1 This answer is redundant with better written, higher upvoted answers, that say essentially the same thing, even though this may have been written earlier. Cest la vie. – toolbear Aug 7 '14 at 19:35

▲
29
▼

I just encountering this for configuration files. I don't want to use **XML** (verbose, graphically, ugly, hard to read), or "ini" format (no hierarchy, no real standard, etc.) or Java "Properties" format (like .ini).

JSON can do all they can do, but it is way less verbose and more human readable - and parsers are easy and ubiquitous in many languages. It's just a tree of data. But out-of-band comments are a necessity often to document "default" configurations and the like. Configurations are never to be "full documents", but trees of saved data that can be human readable when needed.

I guess one could use `"#": "comment"` , for "valid" JSON.

edited Jan 19 '17 at 17:42
Peter Mortensen
14.5k 19 89 118

answered Jun 22 '11 at 13:09
peterk
2,629 4 22 30

4 For config files, I'd suggest YAML, not JSON. It's (almost) a more powerful superset of JSON, but supports more readable constructs as well, including comments. – Marnen Laibow-Koser Oct 19 '11 at 21:35

library dependency. Looks like Ruby 1.9.2 does. Anyone know of others? And which languages ship support for json by default? – nealmcb Mar 21 '12 at 15:53

5 YAML interop is a lie: stackoverflow.com/questions/450399/… . If your instinct is to use JSON for configuration files, follow it. – toolbear Aug 7 '14 at 19:10

This is old, but I believe that using # is not a good idea. Json is close to the syntax of a Javascript litteral. Javascript supports 2 types of comment : // and /* ... */ If I were you I would stick with one or both these types of comments. – Pascal Ganaye Dec 2 '15 at 18:24

28

JSON does not support comments natively, but you can make your own decoder or at least preprocessor to strip out comments, that's perfectly fine (as long as you just ignore comments and don't use them to guide how your application should process the JSON data).

> JSON does not have comments. A JSON encoder MUST NOT output comments. A JSON decoder MAY accept and ignore comments.
>
> Comments should never be used to transmit anything meaningful. That is what JSON is for.

Cf: Douglas Crockford, author of JSON spec.

answered Jun 25 '13 at 14:48
gaborous
9,011 5 53 87

3 Crockford later went on to write: "Suppose you are using JSON to keep configuration files, which you would like to annotate. Go ahead and insert all the comments you like. Then pipe it through JSMin before handing it to your JSON parser." See @kyle-simpson's answer about JSON.minify for more info. – toolbear Aug 7 '14 at 19:14

28

It depends on your JSON library. Json.NET supports JavaScript-style comments, `/* commment */` .

See another Stack Overflow question.

edited May 23 '17 at 10:31
Community ♦
1 1

answered Aug 4 '12 at 0:56
AZ.
4,488 5 36 58

blogs.msdn.com/b/webdev/archive/2014/06/03/… although I haven't found anything in the spec yet. – webXL Sep 17 '14 at 21:54

25

JSON makes a lot of sense for config files and other local usage because it's ubiquitous and because it's much simpler than XML.

If people have strong reasons against having comments in JSON when communicating data (whether valid or not), then possibly JSON could be split into two:

- JSON-COM: JSON on the wire, or rules that apply when communicating JSON data.
- JSON-DOC: JSON document, or JSON in files or locally. Rules that define a valid JSON document.

JSON-DOC will allow comments, and other minor differences might exist such as handling whitespace. Parsers can easily convert from one spec to the other.

With regards to the remark made by Douglas Crockford on this issues (referenced by @Artur Czajka)

> Suppose you are using JSON to keep configuration files, which you would like to annotate. Go ahead and insert all the comments you like. Then pipe it through JSMin before handing it to your JSON parser.

We're talking about a generic config file issue (cross language/platform), and he's answering with a JS specific utility!

Sure a JSON specific minify can be implemented in any language, but standardize this so it becomes ubiquitous across parsers in all languages and platforms so people stop wasting their time lacking the feature because they have good use-cases for it, looking the issue up in online forums, and getting people telling them it's a bad idea or suggesting it's easy to implement stripping comments out of text files.

The other issue is interoperability. Suppose you have a library or API or any kind of subsystem which has some config or data files associated with it. And this subsystem is to be accessed from different languages. Then do you go about telling people: by the way don't forget to strip out the comments from the JSON files before passing them to the parser!

edited Jan 25 '13 at 14:45

answered Dec 11 '12 at 1:37

Basel Shishani

3,796 3 36 57

No need to fragment JSON. JSON with comments is no longer JSON. But it's perfectly acceptable to annotate your JSON with comments, so long as you make sure to strip them out before parsing or transmitting it. It should never be the receiver's responsibility to do this. – toolbear Aug 7 '14 at 19:19

22

The Dojo Toolkit JavaScript toolkit (at least as of version 1.4), allows you to include comments in your JSON. The comments can be of `/* */` format. Dojo Toolkit consumes the JSON via the `dojo.xhrGet()` call.

Other JavaScript toolkits may work similarly.

This can be helpful when experimenting with alternate data structures (or even data lists) before choosing a final option.

edited Jan 19 '17 at 17:39
Peter Mortensen
14.5k 19 89 118

answered Jan 18 '11 at 21:57
David
1,290 1 17 33

4 No. Not this. JSON doesn't have comments. If you choose to annotate your JSON with comments, minify it before parsing or transmitting. This shouldn't be the receiver's responsibility. – toolbear Aug 7 '14 at 19:31

1 I didn't say that JSON has comments. Neither did I mean to imply that it's appropriate to include them in your JSON, especially in a production system. I said that the *Dojo toolkit* permits you to add them, which is (or at least, was) factually true. There are very helpful use-cases out there for doing so in your testing phase. – David Aug 7 '14 at 23:37

1 It's bad voodoo to serve up commented, and thus invalid JSON, which `dojo.xhrGet()` implicitly encourages by accepting. – toolbear Aug 8 '14 at 20:21

I still vote for upgrading the JSON spec to allow comments. I'm all for minifying and stripping the comments before transmitting the JSON, but not having any ability to comment your JSON in any standard way without having to pass it through a separate utility before parsing it just seems silly. I also makes it impossible to use a JSON editor on your JSON configuration files, because your files are not valid JSON. – Craig Sep 15 '14 at 7:03

22

If you use JSON5 you can include comments.

JSON5 is a proposed extension to JSON that aims to make it easier for humans to write and maintain by hand. It does this by adding some minimal syntax features directly from ECMAScript 5.

edited Jan 19 '17 at 18:32
Peter Mortensen
14.5k 19 89 118

answered Nov 24 '15 at 4:34
Smit Johnth
819 11 12

4 Could you please add an example? Then you may actually need those extra characters. – dgilperez Dec 17 '15 at 0:15

stackoverflow.com/help/how-to-answer – dgilperez Dec 29 '15 at 20:06

2 SO is moderated by its users. That means I can provide an answer if I have it the same way I can comment yours if it doesn't follow guidelines. That's how SO gets to be a great resource. – dgilperez Dec 30 '15 at 11:21

21

JSON used to support comments, but they were abused and removed from the standard.

From the creator of JSON:

> I removed comments from JSON because I saw people were using them to hold parsing directives, a practice which would have destroyed interoperability. I know that the lack of comments makes some people sad, but it shouldn't. - Douglas Crockford, 2012

The official JSON site is at JSON.org. JSON is defined as a standard by ECMA International. There is always a petition process to have standards revised. It is unlikely that annotations will be added to the JSON standard for several reasons.

JSON by design is an easily reverse-engineered (human parsed) alternative to XML. It is simplified even to the point that annotations are unnecessary. It is not even a markup language. The goal is stability and interoperablilty.

Anyone who understands the "has-a" relationship of object orientation can understand any JSON structure - that is the whole point. It is just a directed acyclic graph (DAG) with node tags (key/value pairs), which is a near universal data structure.

This only annotation required might be "//These are DAG tags". The key names can be as informative as required, allowing arbitrary semantic arity.

Any platform can parse JSON with just a few lines of code. XML requires complex OO libraries that are not viable on many platforms.

Annotations would just make JSON make less interoperable. There is simply nothing else to add, unless what you really need is a markup language (XML), and don't care if your persisted data is easily parsed.

edited Jun 3 at 2:05

answered Nov 27 '15 at 19:35

Dominic Cerisano

1,756 19 35

2 IMHO, the need for comments in JSON comes from the fact this format is abused for configuration files. We need to have comments in the configuration and therefore using JSON (as defined by standard) for configuration is pretty bad practice. Some parsers allows for line comments but the file then is not strictly JSON format and shouldn't be called JSON. – eXavier Sep 4 '18 at 7:28

dismissive of all the valid reasons I and others are impeded by JSON's lack of comments. – Don Hatch Apr 23 at 17:22

2 You quoted Crockford "I removed comments from JSON because I saw people were using them to hold parsing directives, a practice which would have destroyed interoperability." Do you, or anyone else, understand what this means? How can a comment hold a parsing directive, for a parser that simply ignores comments? – Don Hatch Apr 23 at 17:26

You just answered your own question. – Dominic Cerisano Aug 1 at 20:16

@Dominic Cerisano Don Hatch meant, what's the difference in the comments (text, code, directives, etc) if the interpreter ignores them? Interpreter must ignore all of this in comments, so phrase "I removed comments from JSON because I saw people were using them to hold parsing directives, a practice which would have destroyed interoperability." does not make sense. Now I understand why XML is more prefer for e.g. configuration files. If you have difficult configuration XML is more prefer since you can document it in code (also you can use XML schemes for validate configs). – Alexander Symonenko Sep 11 at 9:43

19 **JSON is not a framed protocol**. It is a *language free format*. So a comment's format is not defined for JSON.

As many people have suggested, there are some tricks, for example, duplicate keys or a specific key `_comment` that you can use. It's up to you.

edited Jan 19 '17 at 18:24
Peter Mortensen
14.5k 19 89 118

answered Jul 20 '15 at 9:26
Manish Shrivastava
20.2k 13 84 98

18 You *can* have comments in [JSONP](), but not in pure JSON. I've just spent an hour trying to make my program work with this example from Highcharts: [http://www.highcharts.com/samples/data/jsonp.php?filename=aapl-c.json&callback=?]()

If you follow the link, you will see

```
?(/* AAPL historical OHLC data from the Google Finance API */
[
/* May 2006 */
[1147651200000,67.79],
[1147737600000,64.98],
...
[1368057600000,456.77],
[1368144000000,452.97]
]);
```

Since I had a similar file in my local folder, there were no issues with the Same-origin policy, so I decided to use pure JSON... and, of course, `$.getJSON` was failing silently because of the comments.

Eventually I just sent a manual HTTP request to the address above and realized that the content-type was `text/javascript` since, well, JSONP returns pure JavaScript. In this case comments *are allowed*. But my application returned content-type `application/json`, so I had to remove the comments.

edited Jan 19 '17 at 18:00
Peter Mortensen
14.5k 19 89 118

answered Oct 7 '13 at 20:37
osa
4,869 35 44

17

This is a **"can you"** question. And here is a **"yes"** answer.

No, you shouldn't use duplicative object members to stuff side channel data into a JSON encoding. (See "The names within an object SHOULD be unique" in the RFC).

And yes, you could insert comments *around* the JSON, which you could parse out.

But if you want a way of inserting and extracting arbitrary side-channel data to a valid JSON, here is an answer. We take advantage of the non-unique representation of data in a JSON encoding. This is allowed[*] in section two of the RFC under "whitespace is allowed before or after any of the six structural characters".

[*] *The RFC only states "whitespace is allowed before or after any of the six structural characters", not explicitly mentioning strings, numbers, "false", "true", and "null". This omission is ignored in ALL implementations.*

First, canonicalize your JSON by minifying it:

```
$jsonMin = json_encode(json_decode($json));
```

Then encode your comment in binary:

```
$hex = unpack('H*', $comment);
$commentBinary = base_convert($hex[1], 16, 2);
```

```
$steg = str_replace('0', ' ', $commentBinary);
$steg = str_replace('1', "\t", $steg);
```

Here is your output:

```
$jsonWithComment = $steg . $jsonMin;
```

edited Dec 15 '15 at 5:53

answered Apr 24 '14 at 17:23
William Entriken
22.1k 13 98 150

1 The RFC only states "whitespace is allowed before or after any of the six structural characters", not explicitly mentioning strings, numbers, "false", "true", "null". This omission is ignored in ALL implementations. – William Entriken Sep 24 '14 at 18:15

1 For greater comment density, couldn't you encode your comment in ternary and use space, tab, and newline to steg it? – Claire Nielsen Sep 26 '18 at 19:44

SHOULD is not MUST. See the explicitly included RFC 2119: MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification. ... SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. – Jeff K Sep 23 at 18:04

Good reference. A better reasoning against using duplicated keys is the standard's quote "When the names within an object are not unique, the behavior of software that receives such an object is unpredictable.". Also now I understand why the standard was not "MUST be unique," this makes a validator simpler, it only needs to track [and {, it does not need to know which keys were used already. – William Entriken Sep 26 at 20:22

12

We are using `strip-json-comments` for our project. It supports something like:

```
/*
 * Description
*/
{
    // rainbows
    "unicorn": /* ❤ */ "cake"
}
```

```
var strip_json_comments = require('strip-json-comments')
var json = '{/*rainbows*/"unicorn":"cake"}';
JSON.parse(strip_json_comments(json));
//=> {unicorn: 'cake'}
```

answered Nov 27 '14 at 11:39

Joy
7,253 6 31 73

Note that the `json` is not a valid JSON anymore when it includes these propriety comments. – Roy Prins Jul 31 at 12:12

12

DISCLAIMER: THIS IS SILLY

There is actually a way to add comments, and stay within the spec (no additional parser needed). It will not result into human-readable comments without any sort of parsing though.

You could abuse the following:

> Insignificant whitespace is allowed before or after any token. Whitespace is any sequence of one or more of the following code points: character tabulation (U+0009), line feed (U+000A), carriage return (U+000D), and space (U+0020).

In a hacky way, you can abuse this to add a comment. For instance: start and end your comment with a tab. Encode the comment in base3 and use the other whitespace characters to represent them. For instance.

```
010212 010202 011000 011000 011010 001012 010122 010121 011021 010202 001012 011022
010212 011020 010202 010202
```

(`hello base three` in ASCII) But instead of 0 use space, for 1 use line feed and for 2 use carriage return.

This will just leave you with a lot of unreadable whitespace (unless you make an IDE plugin to encode/decode it on the fly).

I never even tried this, for obvious reasons and neither should you.

answered Jun 17 at 7:18

11

To cut a JSON item into parts I add "dummy comment" lines:

```
{

"##############################" : "Part1",

"data1"              : "value1",
"data2"              : "value2",

"##############################" : "Part2",

"data4"              : "value3",
"data3"              : "value4"

}
```

edited Jan 19 '17 at 18:01
Peter Mortensen
14.5k 19 89 118

answered Oct 29 '13 at 10:30
Chris
852 10 14

14 You've emulated an INI file structure in JSON. Please, put down your Golden Hammer. – Artur Czajka Nov 18 '13 at 16:53

4 RFC says "The names within an object SHOULD be unique". Also see this person that is having an error parsing JSON like the above: stackoverflow.com/questions/4912386/… – William Entriken Jun 10 '14 at 15:58

If you're using a schema to validate the JSON, it may fail due to the extra fields. – gregsdennis Jun 26 '15 at 0:32

1 If you're really determined to add comments to your JSON, it would make much more sense to do something like this: `{ "comment-001":"This is where you do abc...", "comment-002":"This is where you do xyz..." }` This keeps the name unique and lets you add whatever string value you like. It's still a kludge, because comments should not be part of your JSON. As another alternative, why not add comments before or after your JSON, but not within it? – Jazimov Sep 4 '15 at 22:51

10

JSON doesn't allow comments, per se. The reasoning is utterly foolish, because you can use JSON *itself* to create comments, which obviates the reasoning entirely, *and* loads the parser data space for no good reason at all for *exactly* the same result and potential issues, such as they are: a JSON file with comments.

> **If you try to put comments in (using `//` or `/* */` or `#` for instance), then some parsers will fail because this is strictly not within the JSON specification. So you should *never* do that.**

Here's a for instance, where my image manipulation system has saved image notations and some basic formatted (comment) information relating to them (at the bottom):

```
{
    "Notations": [
        {
            "anchorX": 333,
            "anchorY": 265,
            "areaMode": "Ellipse",
            "extentX": 356,
            "extentY": 294,
            "opacity": 0.5,
            "text": "Elliptical area on top",
            "textX": 333,
            "textY": 265,
            "title": "Notation 1"
        },
        {
            "anchorX": 87,
            "anchorY": 385,
            "areaMode": "Rectangle",
            "extentX": 109,
            "extentY": 412,
            "opacity": 0.5,
            "text": "Rect area\non bottom",
            "textX": 98,
            "textY": 385,
            "title": "Notation 2"
        },
        {
            "anchorX": 69,
            "anchorY": 104,
            "areaMode": "Polygon",
            "extentX": 102,
            "extentY": 136,
            "opacity": 0.5,
            "pointList": [
                {
                    "i": 0,
                    "x": 83,
                    "i": 1,
                    "x": 69,
                    "y": 136
                },
                {
                    "i": 2,
                    "x": 102,
                    "y": 132
                },
                {
                    "i": 3,
                    "x": 83,
                    "y": 104
                }
            ],
            "text": "Simple polygon",
            "textX": 85,
            "textY": 104,
            "title": "Notation 3"
        }
    ],
    "imageXW": 512,
    "imageYW": 512,
    "imageName": "lena_std.ato",
    "tinyDocs": {
        "c01": "JSON image notation data:",
        "c02": "-------------------------",
        "c03": "",
        "c04": "This data contains image notations and related area",
        "c05": "selection information that provides a means for an",
        "c06": "image gallery to display notations with elliptical,",
        "c07": "rectangular, polygonal or freehand area indications",
        "c08": "over an image displayed to a gallery visitor.",
        "c09": "",
        "c10": "X and Y positions are all in image space. The image",
        "c11": "resolution is given as imageXW and imageYW, which",
        "c12": "you use to scale the notation areas to their proper",
        "c13": "locations and sizes for your display of the image,",
        "c14": "regardless of scale.",
        "c15": "",
        "c16": "For Ellipses, anchor is the  center of the ellipse,",
        "c17": "and the extents are the X and Y radii respectively.",
        "c18": "",
        "c19": "For Rectangles, the anchor is the top left and the",
        "c20": "extents are the bottom right.",
        "c21": "",
        "c25": "first, so all you have to be concerned with is drawing",
        "c26": "lines between the points in the list. Anchor and extent",
        "c27": "are set to the top left and bottom right of the indicated",
        "c28": "region, and can be used as a simplistic rectangular",
        "c29": "detect for the mouse hover position over these types",
        "c30": "of areas.",
        "c31": "",
        "c32": "The textx and texty positions provide basic positioning",
        "c33": "information to help you locate the text information",
        "c34": "in a reasonable location associated with the area",
        "c35": "indication.",
        "c36": "",
        "c37": "Opacity is a value between 0 and 1, where .5 represents",
        "c38": "a 50% opaque backdrop and 1.0 represents a fully opaque",
        "c39": "backdrop. Recommendation is that regions be drawn",
        "c40": "only if the user hovers the pointer over the image,",
        "c41": "and that the text associated with the regions be drawn",
        "c42": "only if the user hovers the pointer over the indicated",
        "c43": "region."
    }
}
```

edited Apr 24 at 13:51

answered Jun 19 '18 at 14:04

fyngyrz
1,000 20 31

The "reasoning" link is broken. Any chance of finding a current link to it? – Don Hatch Apr 23 at 17:05

Don, unfortunately, Google has killed the social media system that contained the post; I have no idea where the original poster went from there, if anywhere. I'll kill the link in the above info, though, so as to remove the ambiguity. Thanks. – fyngyrz Apr 24 at 13:50

The reasoning is *not* foolish, and you just proved it. Implementing comments as tags preserves *interoperability*. This is *exactly* why Crockford wanted comments to be parsed as tags. Now *everything* is just a tag and parsed *the same way*. – Dominic Cerisano Jul 14 at 16:30

10

There is a good solution (hack), which is valid JSON, BUT IT WILL NOT WORK IN ALL CASES (see comments bellow). Just make the same key twice (or more). For example:

```
{
  "param" : "This is the comment place",
```

So JSON will understand this as:

```
{
  "param" : "This is value place",
}
```

edited Sep 27 at 20:32

answered Jan 28 '14 at 15:04
Aurimas
506 4 11

8 This method may cause some troubles if anybody will loop through the object. On the first iteration the program will have no information that the entry is a comment. – user2073253 Feb 12 '14 at 20:24

5 RFC says: "The names within an object SHOULD be unique". See this error reported at: stackoverflow.com/questions/4912386/… – William Entriken Jun 10 '14 at 15:59

6 Doing this is an invitation for creating JSON that blows up on you at some random point in the future. – toolbear Aug 7 '14 at 19:00

5 There is no guarantee that order matters in the list of object name/value pairs. A parser could parse them "out of order" and then this is broken. – Mark Lakata May 26 '15 at 22:26

2 Behaviour of a JSON parser with this kind of code is undefined. There is nothing to say that the parser behaves as if only the last value was present. It could behave as if only the first value was present, or any value, or as if the value was an array. – gnasher729 Oct 6 '17 at 18:24

1 2 next

protected by Ajay S Sep 8 '17 at 5:10

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?