JSONC and JSON Lines: Two Plain-Text Variations on JSON

Posted on 2026-11-21

In Why JSON Matters, we looked at how JSON gives plain text a structure that people can read and programs can use. Standard JSON is strict on purpose: no comments, no trailing commas, and one value per file. [1]

Two variations relax those rules for specific jobs. **JSONC** adds comments, for files that people edit by hand. **JSON Lines** puts one record on each line, for data that keeps growing. Both are still plain text.

JSONC: JSON for files people edit

JSONC stands for JSON with Comments. It is JSON plus two kinds of comment: single-line comments that start with `//`, and block comments between `/*` and `*/`. [2]

```
{
  // How many problems each workout contains
  "workoutLength": 5,

  /* Hints are off by default.
     Turn them on for new users. */
  "showHints": false
}
```

The comments explain why a setting has its value. Without them, that explanation has to live in a separate document that can fall out of date.

The rules

- **Comments go where spaces could go.** They can sit between values, but never inside a string, and they never change the data. [2]
- **No nested block comments.** The first `*/` ends the comment. [2]
- **No `#` comments.** JSONC stays a subset of JavaScript syntax, which does not use them. [2]
- **Trailing commas are optional.** A parser may accept a comma after the last item, but it does not have to. Do not rely on them if the file must work with more than one program. [2]

Where you will see it

JSONC is mostly used for configuration. Visual Studio Code keeps its `settings.json`, `launch.json` and `tasks.json` files in JSONC, [2] and the Deno runtime reads its own `deno.jsonc` settings file. [3]

Things to watch

- **Use the `.jsonc` extension.** The JSONC specification recommends it, so people and programs know what to expect. [2] As VS Code shows, some JSONC files still use `.json`, which is why a file that looks like JSON can be rejected by a strict JSON tool.
- **A standard JSON parser will reject it.** As with any format, renaming the file is not enough; the program has to support JSONC.
- **Comments may not survive a rewrite.** A program that reads a JSONC file and saves it again can drop the comments. Keep hand-edited JSONC files hand-edited.
- **The specification is still a draft.** It defines JSONC as what the `jsonc-parser` library accepts with its default settings. [2]

JSON Lines: one record per line

JSON Lines, also called newline-delimited JSON and saved with a `.jsonl` extension, stores a list of records with each record on its own line. [4]

```
{"date": "2026-11-20", "problem": "47 × 18", "answer": 846, "correct": true}
{"date": "2026-11-20", "problem": "63 × 12", "answer": 746, "correct": false}
{"date": "2026-11-21", "problem": "29 × 31", "answer": 899, "correct": true}
```

Each line is a complete JSON value on its own. The file as a whole is not a JSON array: there are no surrounding brackets and no commas between records.

The rules

- **UTF-8 text,** without a byte order mark. [4]
- **Every line is a valid JSON value.** Objects are most common, but any value is allowed. A blank line is not. [4]
- **Lines end with a newline** (`\n`). A newline after the last record is not required but is strongly recommended. [4]

Why one record per line helps

- **Adding is cheap.** A new record goes on the end of the file. With a JSON array, a program has to rewrite the closing bracket, and usually the whole file.
- **Large files stay manageable.** A program can read one line, process it, and move on, instead of loading the whole file into memory first.
- **Damage stays local.** If one line is cut off or mistyped, the other lines are still readable.
- **Everyday text tools work.** You can count records by counting lines, or search for a record with a plain text search.

That makes JSON Lines a common choice for logs, exports, history files and datasets that grow over time.

Things to watch

- **No pretty printing.** Each record must stay on one line, so long records are harder to read by eye than indented JSON.
- **A standard JSON parser will reject the whole file,** because it contains more than one value. Read it one line at a time instead.
- **NDJSON is a close cousin.** It follows a similar idea but also allows blank lines, so a tool built for one may not accept every file made for the other. [5]

Which one to use

Format	Best for	Comments	Shape of the file
JSON (`.json`)	Exchanging data between programs	No	One value, often an object or array
JSONC (`.jsonc`)	Settings files people edit by hand	Yes	One value, with comments
JSON Lines (`.jsonl`)	Logs, history and large datasets	No	One value per line

A simple rule: if a person edits the file, consider JSONC. If a program keeps adding records to it, consider JSON Lines. If other programs need to read it, standard JSON is the safest choice.

To open all three on Android, Plain JSON is our read-only viewer for JSON, JSONC and JSON Lines. It shows each file as a tree and never changes it.

Sources

1. Bray T. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC Editor; 2017. RFC 8259. RFC 8259. DOI: 10.17487/RFC8259.
2. *JSONC Specification* [Internet]. Accessed 2026-11-21. https://jsonc.org/.
3. *Parsing JSONC*. Deno Docs [Internet]. Accessed 2026-11-21. https://docs.deno.com/examples/parsing_jsonc/.
4. *JSON Lines* [Internet]. Accessed 2026-11-21. https://jsonlines.org/.
5. *JSON Lines: On the web* [Internet]. Accessed 2026-11-21. https://jsonlines.org/on_the_web.