jaytail manual

Version 0.2.2

Contents

Usage 1
Input 2
Columns 2
Search and filters 2
Selection, copy and export 3
Browser mode 4
Sending a log to someone 4
Asking someone for a log 5
Examples 8
Limits 8
License 8
Security and privacy 8

jaytail is a small, fast, read-only pager for JSON Lines logs (one JSON object per line, also called JSONL or NDJSON). It opens a file, a pipe, or a file picker, and shows each record as a table row with automatically recognized columns.

jaytail never modifies its input.

Usage

```
jaytail FILE                open FILE in the terminal viewer
jaytail                     no FILE and an interactive terminal: open the file picker
cmd | jaytail               no FILE and piped input: read standard input
jaytail -                   read standard input explicitly
jaytail FILE | cmd          stdout is not a terminal: copy records as raw JSONL
jaytail --plain=table FILE  print an aligned text table without escape sequences
jaytail man                 show this manual as a man page (roff when redirected)
jaytail manual              show this manual as plain text
jaytail version             print the version (also: ver, vers, --version)
jaytail completions SHELL   print a completion script: bash, zsh, fish, powershell, elvish
```

Options:

```
--columns LIST      initial columns, comma separated: time, level, message or
                    field paths such as service or request.id
--level LEVEL       initial level filter: `warn` shows WARN and more severe
                    records, `=warn` shows only WARN records
--time-format FMT   strftime pattern of the time column
                    (default "%Y-%m-%d %H:%M:%S %z")
--utc               show times in UTC instead of the local timezone
--no-color          no colors or text styles (NO_COLOR is honored too)
--plain[=FORMAT]    noninteractive output: jsonl (default) or table
--web               serve a browser UI on 127.0.0.1 only
--no-open, --port N web mode only: do not launch a browser / choose the port
--help, --version
```

Aliases: `jtail` and `jt` run jaytail too. The installers create them only when the name does not belong to another program (and say so when they skip one).

Shell completion: `jaytail completions SHELL` prints the script (`--name jtail` or `--name jt` for an alias). install.sh and `make install` put the bash, zsh and fish scripts under ~/.local/share (zsh: add ~/.local/share/zsh/site-functions to fpath). PowerShell: add `jaytail completions powershell | Out-String | Invoke-Expression` to $PROFILE.

Exit status: 0 on success, 1 on runtime errors (unreadable input, terminal failure), 2 on invalid arguments, 3 when a license is required (see LICENSE).

Input

Each physical line is one record. Lines may end with LF or CRLF; the last line may lack a terminator; a UTF-8 byte order mark at the start of the file is accepted. Lines that are not JSON objects stay visible: valid JSON values that are not objects are marked NON-OBJECT, unparsable lines (including invalid UTF-8) are marked INVALID JSON. Lines longer than 8 MiB are not parsed: they are shown as OVERSIZED with a preview but remain part of the file.

Regular files are opened as a snapshot: data appended later is not shown (jaytail reports that the file grew; reopen it with `o`). If the file is truncated or replaced, jaytail reports that the view may be stale.

Piped input is copied to a private temporary file that is deleted on exit, so you can scroll back through everything received so far. Only complete lines are shown until the input ends.

Columns

jaytail samples up to 64 records from the start, 64 from the end and 128 from the middle of a file (from a pipe: the first 256 objects or 1 MiB) and picks the default columns in this order:

```
TIME       first of: time, ts, timestamp, @timestamp, datetime, date
LEVEL      first of: level, lvl, severity, severity_text, loglevel, log.level
component, method, url    when present, in this order
MESSAGE    first of: message, msg, text, event
others     the most common short fields (service, caller, request_id, ...)
```

Key names are matched case-insensitively. Times are shown with the --time-format pattern in the local timezone (--utc for UTC). RFC 3339 strings and Unix epochs in seconds (also fractional), milliseconds, microseconds or nanoseconds are recognized; values that are not plausible timestamps are shown unchanged.

Levels are normalized to TRACE, DEBUG, INFO, WARN, ERROR, FATAL, PANIC. Accepted spellings: trace/trc, debug/dbg, info/inf/msg, warn/warning/wrn, error/err, fatal/ftl/critical/crit, panic, and the numeric levels 10-60 used by pino and bunyan. Other values are shown as-is.

Press `R` to re-sample the file and recompute the columns.

Search and filters

Search (`/` or Ctrl+F) highlights matching rows without hiding others:

```
timeout                case-insensitive text anywhere in the original line,
                       including keys and fields not shown as columns
service:payments       the decoded value of field `service` contains "payments"
request.id:abc         nested field request → id
request\.id:abc        top-level key "request.id" (escaped dot)
http\:500              the literal text "http:500" (escaped colon)
message:refused        `time`, `level` and `message` also resolve their aliases
```

The first unescaped colon separates a field path from the text, unless the part before it contains spaces. `\.`, `\:` and `\\` escape a dot, a colon and a backslash. Malformed lines match text searches but never field searches. Queries are limited to 4 KiB and 16 nested path components.

n / N (or F3 / Shift+F3) move to the next / previous match in file order, among the rows that are currently visible. M shows only matching rows.

Filters hide rows; they combine with AND:

```
level        1 = all, 2 = WARN and above, 3 = ERROR and above, L = choose,
             = switches between "at least" and "exactly";
             --level warn / --level =warn on the command line.
             Records without a recognized level are hidden by any level filter.
field        f, then field:text (case-insensitive substring of the decoded
             value; missing or null fields do not match). Empty input clears.
```

Filtered rows keep their order and physical line numbers. The status bar shows the number of matching rows once the whole file has been checked ("≥N rows" while counting). Ctrl+G to a line hidden by a filter reports it instead of jumping elsewhere.

Selection, copy and export

Selections are made of record identities (file offsets), never copies of rows:

```
Space         mark or unmark the focused row
v             start a range at the focus; move; v again to select it
              (Shift+Up/Down also extend a range). A range made while a
              filter is active covers only the rows visible under it.
Ctrl+A        select every row of the current view. The filter and the
              bytes available at that moment are frozen: changing filters
              later does not change this selection.
u             clear the selection (asks first when it is large or still
              being counted)
```

The status bar shows "N selected" once the count is known ("≥N" while counting). Unmarking a row inside a range or Ctrl+A selection excludes it. Marks are limited to 100,000 individual rows; use ranges or Ctrl+A beyond that.

y copies the selected lines (or the focused line when nothing is selected) as original text, in file order, one per line, with a final newline. The limit is 10 MiB; larger selections must be exported. jaytail uses the system clipboard; in SSH sessions or when no clipboard is available it sends an OSC 52 request to the terminal (up to 100 KiB) and says that delivery is not confirmed, because many terminals ignore or limit it. Invalid UTF-8 is replaced and reported.

e opens the export dialog: scope (selected rows, or all rows of the current view), format, columns (visible, or all discovered fields) and output path. The default path is <name>.selection.<ext> next to the source (or in the current directory if that is not writable; jaytail-export.<ext> for piped input).

```
JSONL   original lines byte for byte (line endings become LF); malformed
        lines are kept as they are
JSON    one array; valid records keep their original text, other lines
        become {"_jaytail":{"status":"invalid_json","line":N,"byte_offset":B},
        "raw":"..."} with "raw_base64" when the bytes are not valid UTF-8;
        lines over 8 MiB become {"_jaytail":{"status":"oversized",
        "raw_unavailable":true,...},"preview":"..."} (use JSONL to keep them)
CSV     header from the column labels plus "raw" and "status"; original
        values (no time formatting); objects and arrays as compact JSON;
        strings starting with = + - @ | % (or a tab/CR/LF) get a leading '
```

```
        so spreadsheets do not run them as formulas
HTML    one standalone page without scripts or remote resources; cells show
        the same text as the table and each row keeps its raw event
```

Exports stream to a temporary file in the target directory that is renamed into place only when complete; cancelling (Esc) or an error removes it. An existing file is replaced only after confirmation, and the source file is never overwritten.

Browser mode

```
jaytail --web FILE            serve FILE and open the default browser
jaytail --web --no-open FILE  only print the URL
jaytail --web --port 8080 FILE
```

The server listens on 127.0.0.1 only (never on other interfaces) and prints a URL containing a random 256-bit session token, for example http://127.0.0.1:53211/s/<token>/. Anyone who can reach that URL on this machine can read the log, so do not share it. Requests with another Host or Origin, or without the token, are rejected. The browser cannot choose files: without FILE the file picker runs in the terminal first. Ctrl+C in the terminal stops the server; closing the browser does not.

The browser shows the same columns, filters, search, selection, copy and export as the terminal (see the Browser key table below). Copy uses the browser clipboard (it may ask for permission; when it is refused the text is shown for manual copying). Exports are downloaded by the browser.

Sending a log to someone

```
jaytail send FILE                  offer FILE and print an offer ticket
jaytail receive TICKET             open the file in the viewer while it downloads
jaytail receive TICKET -o PATH     save it to PATH (never overwrites a file)
jaytail receive TICKET > PATH      stdout is not a terminal: write the raw bytes
jaytail send FILE --qr             also show the ticket as a QR code
jaytail receive --qr-image PHOTO   read the ticket from a photo of that QR code
```

`jaytail send` prints a ticket (a long string starting with jaytail1) and waits for one receiver. Give the ticket to the other person over a channel you trust: whoever has it can fetch the file while `jaytail send` runs. The offer ends after the first complete transfer, after --expire (default 15m), or with Ctrl+C. The file is sent as it was when `jaytail send` started; if it is truncated or replaced meanwhile, the transfer fails.

Under the ticket, jaytail prints a link (https://jaytail.dev/t#jaytail1...) for the other person: the page shows how to get jaytail, with the installer or as a verified download without installing, and the command to run with the ticket filled in. The ticket is in the part after #, which the browser never sends to the server. With jaytail installed, the short command is enough; `receive` and `send` also accept the link itself in place of the ticket.

The connection is encrypted end to end and goes directly between the two machines when possible, otherwise through the jaytail relay (relay.jaytail.dev), which forwards encrypted packets and cannot read them. The receiver verifies a BLAKE3 checksum of the whole file; a mismatch is an error and -o leaves no file.

If the connection drops, the side that answered the ticket reconnects and continues where it stopped (up to 5 times in a row, giving up after 2 minutes without progress); the checksum still covers the whole file. If a direct path stays below 1 MiB/s for 10 seconds, that side tries the relay and keeps it only when it is at least 1.5 times faster; it says so on the terminal.

Asking someone for a log

```
jaytail request                    print a request ticket; files go to the current directory
jaytail request -o DIR             save the files into DIR
jaytail send TICKET FILE           on the other machine: send FILE to the requester
jaytail send --qr-image PHOTO FILE the same, with the ticket read from a photo of its QR
code
```

When you support a system you cannot log into, ask for the log file instead of having it pasted into a chat: `jaytail request` prints a request ticket and, below it, the link to send to the other person (the page explains how to answer). `jaytail send TICKET FILE` prints the file, its size and the requester's short id before it connects, sends exactly that file, and needs no license and no privileges. It ends when the requester has verified the checksum.

A request stays open for --expire (default 1h) or until Ctrl+C, and accepts files from the first person who answers it; anyone else holding the ticket is refused. That person may run `jaytail send TICKET FILE` again for more files: to be recognized, `send` keeps one random key per ticket in jaytail's data directory (next to the evaluation state) for 7 days. Each file is saved under the sender's file name, made safe for every platform, and never replaces an existing file (app.jsonl, then app-1.jsonl). On a terminal the first file also opens in the viewer as it arrives (--no-view only saves); messages about later files appear when the viewer closes, and the request keeps waiting until it expires or Ctrl+C.

Options of send, receive and request:

```
--relay URL      use another relay (also $JAYTAIL_RELAY); `none` for direct only
--relay-only     never connect directly; a ticket then contains no IP
                 addresses of its creator
--expire DUR     (send FILE, request) how long to wait: 90s, 15m, 2h
--no-auto-relay  (receive, send TICKET FILE) keep a slow direct path instead
                 of trying the relay
--qr             (send FILE, request) draw the link as a QR code below the
                 ticket, when the output is a terminal: a phone camera opens
                 the page, or a photo of it works with --qr-image, when you
                 cannot copy or send the text (--relay-only makes it smaller)
--qr-image FILE  (receive, send) read the ticket from a JPEG or PNG photo or
                 screenshot of such a QR code, instead of TICKET
-o DIR           (request) directory for the received files (default: .)
--no-view        (request) do not open the first file in the viewer
```

A ticket says whether it offers a file or asks for one; `receive` and `send` refuse the other kind and name the right command. A ticket that names a different relay than the configured one is refused unless --relay names that relay. A file literally named send, receive or request is opened as ./send.
Keys

Log viewer

Key	Action
`Up/Down`, `k/j`	Move focused row
`Left/Right`, `h/l`	Scroll columns horizontally
`PageUp/PageDown`, `Ctrl+B`	Move one page
`Home/End`, `g/G`	First/last row
`Ctrl+G`, `:N`	Go to physical line N
`/`, `Ctrl+F`	Search text, or field:text
`n/N`, `F3/Shift+F3`	Next/previous match
`M`	Show only matching rows (toggle)
`f`	Field filter field:text (empty clears)
`1`, `2`, `3`	Level filter: all, WARN+, ERROR+
`L`	Choose the level filter
`=`	Level filter: at least / exactly
`Enter`	Open record detail
`Space`	Mark or unmark the row
`v`	Start/finish a range selection
`Shift+Up/Down`	Extend a range selection
`Ctrl+A`	Select all rows of the current view
`u`	Clear the selection
`y`	Copy selected (or focused) lines
`c`	Column chooser
`e`	Export dialog
`R`	Re-sample the file and refresh default columns
`o`	Open another file
`?`, `F1`	Help
`Esc`	Close prompt; cancel range, scan or export
`q`	Quit

Record detail

Key	Action
`Up/Down`, `k/j`	Scroll
`Left/Right`, `h/l`	Scroll horizontally
`PageUp/PageDown`	Scroll one page
`Home/End`, `g/G`	Top/bottom
`Esc`, `q`	Back to the table

File picker

Key	Action
`Up/Down`, `k/j`	Move cursor
`PageUp/PageDown`	Move one page
`Home/End`, `g/G`	First/last entry
`Enter`	Open directory or file
`Backspace`	Parent directory
`/`	Filter file names
`Ctrl+L`	Type a path
`Esc`, `q`	Quit (or back to the viewer)

Column chooser

Key	Action
`Up/Down`, `k/j`	Move cursor
`Space`	Show or hide the column
`K/J`	Move the column up/down
`Enter`	Apply
`Esc`	Cancel

Browser (--web)

Key	Action
`Up/Down`, `k/j`	Move focused row
`PageUp/PageDown`, `Home/End`	Move one page / first or last row
`Enter`, `double-click`	Record detail
`Space`, `Ctrl/Cmd+click`	Mark or unmark the row
`v`, `Shift+click`, `Shift+Up/Down`	Range selection
`Ctrl+A`	Select all rows of the current view
`u`	Clear the selection
`/`	Search box (Ctrl+F too, when the browser allows it)
`n/N`, `F3/Shift+F3`	Next/previous match
`M`	Show only matching rows (toggle)
`f`	Field filter box
`1`, `2`, `3`, `L`, `=`	Level filter: all, WARN+, ERROR+, choose, mode
`:`, `Ctrl+G`	Go to physical line
`y`	Copy selected (or focused) lines
`e`	Export (download)
`c`	Columns
`?`	Help
`Esc`	Close a panel or cancel a range

Examples

```
zerolog / zap / logrus / slog JSON output:   jaytail service.log
kubectl:     kubectl logs deploy/api | jaytail
journald:    journalctl -o json -u nginx | jaytail
only errors: jaytail --level =error app.jsonl
plain table: jaytail --plain=table --columns time,level,message app.jsonl
```

Limits

- One input per invocation.
- Records are always shown in file order; there is no sorting.
- Line counts are reported only once the background index has finished.
- Lines over 8 MiB are not parsed.

License

```
jaytail license                        status (also --json)
jaytail license show                   every installed license
jaytail license install FILE           install a .license file
jaytail license install --key LICENSE  install a pasted license line (JTL1...)
jaytail license remove LICENSE_ID
```

Licenses are checked offline: no account, no activation, no connection. A license is perpetual for every version released within its update period (Lifetime: all versions). Without one, viewing works for a 7-day evaluation; after that, viewing ends with exit status 3 until a license is installed. Answering a transfer ticket (`jaytail receive TICKET`, `jaytail send TICKET FILE`) is always free; creating one (`jaytail send FILE`, `jaytail request`) needs a Complete license (not available during the evaluation). JAYTAIL_LICENSE can hold a license line for a single run (CI, containers). Licenses: https://jaytail.dev

Security and privacy

Log content is treated as untrusted: control characters, terminal escape sequences and bidirectional-text controls are shown as visible escapes (for example , , <U+202E>) and are never sent to the terminal. jaytail connects to the network only in `jaytail send`, `jaytail receive` and `jaytail request`.