

NAME

loginctl – Control the systemd login manager

SYNOPSIS

loginctl [OPTIONS...] {COMMAND} [NAME...]

DESCRIPTION

loginctl may be used to introspect and control the state of the **systemd**(1) login manager **systemd-logind.service**(8).

OPTIONS

The following options are understood:

--no-ask-password

Do not query the user for authentication for privileged operations.

-p, --property=

When showing session/user/seat properties, limit display to certain properties as specified as argument. If not specified, all set properties are shown. The argument should be a property name, such as "Sessions". If specified more than once, all properties with the specified names are shown.

--value

When showing session/user/seat properties, only print the value, and skip the property name and "=".

-a, --all

When showing session/user/seat properties, show all properties regardless of whether they are set or not.

-l, --full

Do not ellipsize process tree entries.

--kill-who=

When used with **kill-session**, choose which processes to kill. Must be one of **leader**, or **all** to select whether to kill only the leader process of the session or all processes of the session. If omitted, defaults to **all**.

-s, --signal=

When used with **kill-session** or **kill-user**, choose which signal to send to selected processes. Must be one of the well known signal specifiers, such as **SIGTERM**, **SIGINT** or **SIGSTOP**. If omitted, defaults to **SIGTERM**.

-n, --lines=

When used with **user-status** and **session-status**, controls the number of journal lines to show, counting from the most recent ones. Takes a positive integer argument. Defaults to 10.

-o, --output=

When used with **user-status** and **session-status**, controls the formatting of the journal entries that are shown. For the available choices, see **journalctl**(1). Defaults to "short".

-H, --host=

Execute the operation remotely. Specify a hostname, or a username and hostname separated by "@", to connect to. The hostname may optionally be suffixed by a port ssh is listening on, separated by ":", and then a container name, separated by "/", which connects directly to a specific container on the specified host. This will use SSH to talk to the remote machine manager instance. Container names may be enumerated with **machinectl -H HOST**. Put IPv6 addresses in brackets.

-M, --machine=

Execute operation on a local container. Specify a container name to connect to.

--no-pager

Do not pipe output into a pager.

--no-legend

Do not print the legend, i.e. column headers and the footer with hints.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

COMMANDS

The following commands are understood:

Session Commands**list-sessions**

List current sessions.

session-status [ID...]

Show terse runtime status information about one or more sessions, followed by the most recent log data from the journal. Takes one or more session identifiers as parameters. If no session identifiers are passed, the status of the caller's session is shown. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use **show-session** instead.

show-session [ID...]

Show properties of one or more sessions or the manager itself. If no argument is specified, properties of the manager will be shown. If a session ID is specified, properties of the session are shown. By default, empty properties are suppressed. Use **--all** to show those too. To select specific properties to show, use **--property=**. This command is intended to be used whenever computer-parsable output is required. Use **session-status** if you are looking for formatted human-readable output.

activate [ID]

Activate a session. This brings a session into the foreground if another session is currently in the foreground on the respective seat. Takes a session identifier as argument. If no argument is specified, the session of the caller is put into foreground.

lock-session [ID...], unlock-session [ID...]

Activates/deactivates the screen lock on one or more sessions, if the session supports it. Takes one or more session identifiers as arguments. If no argument is specified, the session of the caller is locked/unlocked.

lock-sessions, unlock-sessions

Activates/deactivates the screen lock on all current sessions supporting it.

terminate-session ID...

Terminates a session. This kills all processes of the session and deallocates all resources attached to the session.

kill-session ID...

Send a signal to one or more processes of the session. Use **--kill-who=** to select which process to kill. Use **--signal=** to select the signal to send.

User Commands**list-users**

List currently logged in users.

user-status [USER...]

Show terse runtime status information about one or more logged in users, followed by the most recent log data from the journal. Takes one or more user names or numeric user IDs as parameters. If no parameters are passed, the status is shown for the user of the session of the caller. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use **show-user** instead.

show-user [USER...]

Show properties of one or more users or the manager itself. If no argument is specified, properties of the manager will be shown. If a user is specified, properties of the user are shown. By default, empty properties are suppressed. Use **--all** to show those too. To select specific properties to show, use

--property=. This command is intended to be used whenever computer-parsable output is required. Use **user-status** if you are looking for formatted human-readable output.

enable-linger [*USER...*], **disable-linger** [*USER...*]

Enable/disable user lingering for one or more users. If enabled for a specific user, a user manager is spawned for the user at boot and kept around after logouts. This allows users who are not logged in to run long-running services. Takes one or more user names or numeric UIDs as argument. If no argument is specified, enables/disables lingering for the user of the session of the caller.

See also *KillUserProcesses=* setting in **logind.conf**(5).

terminate-user *USER...*

Terminates all sessions of a user. This kills all processes of all sessions of the user and deallocates all runtime resources attached to the user.

kill-user *USER...*

Send a signal to all processes of a user. Use **--signal=** to select the signal to send.

Seat Commands

list-seats

List currently available seats on the local system.

seat-status [*NAME...*]

Show terse runtime status information about one or more seats. Takes one or more seat names as parameters. If no seat names are passed the status of the caller's session's seat is shown. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use **show-seat** instead.

show-seat [*NAME...*]

Show properties of one or more seats or the manager itself. If no argument is specified, properties of the manager will be shown. If a seat is specified, properties of the seat are shown. By default, empty properties are suppressed. Use **--all** to show those too. To select specific properties to show, use **--property=**. This command is intended to be used whenever computer-parsable output is required. Use **seat-status** if you are looking for formatted human-readable output.

attach *NAME DEVICE...*

Persistently attach one or more devices to a seat. The devices should be specified via device paths in the /sys file system. To create a new seat, attach at least one graphics card to a previously unused seat name. Seat names may consist only of a-z, A-Z, 0-9, "-" and "_" and must be prefixed with "seat". To drop assignment of a device to a specific seat, just reassign it to a different seat, or use **flush-devices**.

flush-devices

Removes all device assignments previously created with **attach**. After this call, only automatically generated seats will remain, and all seat hardware is assigned to them.

terminate-seat *NAME...*

Terminates all sessions on a seat. This kills all processes of all sessions on the seat and deallocates all runtime resources attached to them.

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

EXAMPLES

Example 1. Querying user status

```
$ loginctl user-status
fatima (1005)
  Since: Sat 2016-04-09 14:23:31 EDT; 54min ago
  State: active
  Sessions: 5 *3
  Unit: user-1005.slice
```

```

user@1005.service
...
session-3.scope
...
session-5.scope
3473 login -- fatima
3515 -zsh

```

```

Apr 09 14:40:30 laptop login[2325]: pam_unix(login:session):
    session opened for user fatima by LOGIN(uid=0)
Apr 09 14:40:30 laptop login[2325]: LOGIN ON tty3 BY fatima

```

There are two sessions, 3 and 5. Session 3 is a graphical session, marked with a star. The tree of processing including the two corresponding scope units and the user manager unit are shown.

ENVIRONMENT

\$SYSTEMD_PAGER

Pager to use when **--no-pager** is not given; overrides *\$PAGER*. If neither *\$SYSTEMD_PAGER* nor *\$PAGER* are set, a set of well-known pager implementations are tried in turn, including **less**(1) and **more**(1), until one is found. If no pager implementation is discovered no pager is invoked. Setting this environment variable to an empty string or the value "cat" is equivalent to passing **--no-pager**.

\$SYSTEMD_LESS

Override the options passed to **less** (by default "FRSXMK").

If the value of *\$SYSTEMD_LESS* does not include "K", and the pager that is invoked is **less**, Ctrl+C will be ignored by the executable. This allows **less** to handle Ctrl+C itself.

\$SYSTEMD_LESSCHARSET

Override the charset passed to **less** (by default "utf-8", if the invoking terminal is determined to be UTF-8 compatible).

SEE ALSO

systemd(1), **systemctl**(1), **systemd-logind.service**(8), **logind.conf**(5)