

**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-legend**

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

**--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.

**-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**.

**-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 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**.

**-M, --machine=**

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

**-h, --help**

Print a short help text and exit.

**--version**

Print a short version string and exit.

**--no-pager**

Do not pipe output into a pager.

The following commands are understood:

**list-sessions**

List current sessions.

**session-status** *ID...*

Show terse runtime status information about one or more sessions. 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 one or more sessions. This brings one or more sessions into the foreground, if another session is currently in the foreground on the respective seat.

**lock-session** *ID...*, **unlock-session** *ID...*

Activates/deactivates the screen lock on one or more sessions, if the session supports it.

**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.

**list-users**

List currently logged in users.

**user-status** *USER...*

Show terse runtime status information about one or more logged in users. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use **show-user** instead. Users may be specified by their usernames or numeric user IDs.

**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.

**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.

**list-seats**

List currently available seats on the local system.

**seat-status** *NAME...*

Show terse runtime status information about one or more seats. 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.

**ENVIRONMENT***SYSTEMD\_PAGER*

Pager to use when **--no-pager** is not given; overrides *PAGER*. Setting this to an empty string or the value cat is equivalent to passing **--no-pager**.

*SYSTEMD\_LESS*

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

**SEE ALSO**

[systemd\(1\)](#), [systemctl\(1\)](#), [systemd-logind.service\(8\)](#), [logind.conf\(5\)](#)