

NAME

xcb_input_list_input_devices -

SYNOPSIS

```
#include <xcb/xinput.h>
```

Request function

```
xcb_input_list_input_devices_cookie_t xcb_input_list_input_devices(xcb_connection_t *conn,
```

Reply datastructure

```
typedef struct xcb_input_list_input_devices_reply_t {
    uint8_t response_type;
    uint8_t xi_reply_type;
    uint16_t sequence;
    uint32_t length;
    uint8_t devices_len;
    uint8_t pad0[23];
} xcb_input_list_input_devices_reply_t;
```

Reply function

```
xcb_input_list_input_devices_reply_t *xcb_input_list_input_devices_reply(xcb_connection_t *conn,
    xcb_input_list_input_devices_cookie_t cookie, xcb_generic_error_t **e);
```

Reply accessors

```
xcb_input_device_info_t *xcb_input_list_input_devices_devices(const
    xcb_input_list_input_devices_request_t *reply);
```

```
int xcb_input_list_input_devices_devices_length(const xcb_input_list_input_devices_reply_t *reply);
```

```
xcb_input_device_info_iterator_t xcb_input_list_input_devices_devices_iterator(const
    xcb_input_list_input_devices_reply_t *reply);
```

```
int xcb_input_list_input_devices_infos_length(const xcb_input_list_input_devices_reply_t *reply);
```

```
xcb_input_input_info_iterator_t xcb_input_list_input_devices_infos_iterator(const
    xcb_input_list_input_devices_reply_t *reply);
```

```
int xcb_input_list_input_devices_names_length(const xcb_input_list_input_devices_reply_t *reply);
```

```
xcb_str_iterator_t xcb_input_list_input_devices_names_iterator(const
    xcb_input_list_input_devices_reply_t *reply); uint8_t *xcb_input_list_input_devices_pad_1 (const
    xcb_input_list_input_devices_request_t *reply)
```

REQUEST ARGUMENTS

conn The XCB connection to X11.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_INPUT_LIST_INPUT_DEVICES*. This field is also present in the *xcb_generic_reply_t* and can be used to tell replies apart from each other.

sequence The sequence number of the last request processed by the X11 server.

length The length of the reply, in words (a word is 4 bytes).

xi_reply_type TODO: NOT YET DOCUMENTED.

devices_len TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_input_list_input_devices_cookie_t*. Errors have to be handled when calling the reply function *xcb_input_list_input_devices_reply*.

If you want to handle errors in the event loop instead, use *xcb_input_list_input_devices_unchecked*. See **xcb-requests(3)** for details.

ERRORS

This request does never generate any errors.

SEE ALSO

AUTHOR

Generated from xinput.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.