

NAME

xcb_input_xi_get_client_pointer -

SYNOPSIS

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

Request function

```
xcb_input_xi_get_client_pointer_cookie_t xcb_input_xi_get_client_pointer(xcb_connection_t *conn,
    xcb_window_t window);
```

Reply datastructure

```
typedef struct xcb_input_xi_get_client_pointer_reply_t {
    uint8_t      response_type;
    uint8_t      pad0;
    uint16_t     sequence;
    uint32_t     length;
    uint8_t      set;
    uint8_t      pad1;
    xcb_input_device_id_t deviceid;
    uint8_t      pad2[20];
} xcb_input_xi_get_client_pointer_reply_t;
```

Reply function

```
xcb_input_xi_get_client_pointer_reply_t
    *xcb_input_xi_get_client_pointer_reply(xcb_connection_t *conn,
    xcb_input_xi_get_client_pointer_cookie_t cookie, xcb_generic_error_t **e);
```

REQUEST ARGUMENTS

conn The XCB connection to X11.

window TODO: NOT YET DOCUMENTED.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_INPUT_XI_GET_CLIENT_POINTER*. 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).

set TODO: NOT YET DOCUMENTED.

deviceid TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_input_xi_get_client_pointer_cookie_t*. Errors have to be handled when calling the reply function *xcb_input_xi_get_client_pointer_reply*.

If you want to handle errors in the event loop instead, use *xcb_input_xi_get_client_pointer_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.