

xcb\_input\_get\_device\_modifier\_mapping(3) XCB Requests xcb\_input\_get\_device\_modifier\_mapping(3)

## NAME

xcb\_input\_get\_device\_modifier\_mapping -

## SYNOPSIS

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

### Request function

```
xcb_input_get_device_modifier_mapping_cookie_t  
xcb_input_get_device_modifier_mapping(xcb_connection_t *conn, uint8_t device_id);
```

### Reply datastructure

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

### Reply function

```
xcb_input_get_device_modifier_mapping_reply_t  
*xcb_input_get_device_modifier_mapping_reply(xcb_connection_t *conn,  
xcb_input_get_device_modifier_mapping_cookie_t cookie, xcb_generic_error_t **e);
```

### Reply accessors

```
uint8_t *xcb_input_get_device_modifier_mapping_keymaps(const  
xcb_input_get_device_modifier_mapping_request_t *reply);  
  
int xcb_input_get_device_modifier_mapping_keymaps_length(const  
xcb_input_get_device_modifier_mapping_reply_t *reply);  
  
xcb_generic_iterator_t xcb_input_get_device_modifier_mapping_keymaps_end(const  
xcb_input_get_device_modifier_mapping_reply_t *reply);
```

## REQUEST ARGUMENTS

*conn*           The XCB connection to X11.

*device\_id*      TODO: NOT YET DOCUMENTED.

xcb\_input\_get\_device\_modifier\_mapping(3) XCB Requests xcb\_input\_get\_device\_modifier\_mapping(3)

## REPLY FIELDS

*response\_type* The type of this reply, in this case `XCB_INPUT_GET_DEVICE_MODIFIER_MAPPING`. 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.

*keycodes\_per\_modifier*  
TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

Returns an `xcb_input_get_device_modifier_mapping_cookie_t`. Errors have to be handled when calling the reply function `xcb_input_get_device_modifier_mapping_reply`.

If you want to handle errors in the event loop instead, use `xcb_input_get_device_modifier_mapping_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.