

**NAME**

xcb\_input\_xi\_query\_device -

**SYNOPSIS**

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

**Request function**

```
xcb_input_xi_query_device_cookie_t xcb_input_xi_query_device(xcb_connection_t *conn,
    xcb_input_device_id_t deviceid);
```

**Reply datastructure**

```
typedef struct xcb_input_xi_query_device_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint16_t num_infos;
    uint8_t pad1[22];
} xcb_input_xi_query_device_reply_t;
```

**Reply function**

```
xcb_input_xi_query_device_reply_t *xcb_input_xi_query_device_reply(xcb_connection_t *conn,
    xcb_input_xi_query_device_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
int xcb_input_xi_query_device_infos_length(const xcb_input_xi_query_device_reply_t *reply);
```

```
xcb_input_xi_device_info_iterator_t xcb_input_xi_query_device_infos_iterator(const
    xcb_input_xi_query_device_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*deviceid*       TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

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

*num\_infos*     TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

Returns an *xcb\_input\_xi\_query\_device\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_input\_xi\_query\_device\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_input\_xi\_query\_device\_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](mailto:xcb@lists.freedesktop.org) for corrections and improvements.