NAME

xcb_dri2_connect -

SYNOPSIS

#include <xcb/dri2.h>

Request function

xcb_dri2_connect_cookie_t **xcb_dri2_connect**(xcb_connection_t *_conn_, xcb_window_t _window_, uint32_t _driver_type_);

Reply datastructure

```
typedef struct xcb_dri2_connect_reply_t {
    uint8_t  response_type;
    uint8_t  pad0;
    uint16_t sequence;
    uint32_t length;
    uint32_t driver_name_length;
    uint32_t device_name_length;
    uint8_t  pad1[16];
} xcb_dri2_connect_reply_t;
```

Reply function

xcb_dri2_connect_reply_t *__xcb_dri2_connect_reply__(xcb_connection_t *_conn_, xcb_dri2_connect_cookie_t _cookie_, xcb_generic_error_t **_e_);

Reply accessors

char *__xcb_dri2_connect_driver_name__(const xcb_dri2_connect_request_t *_reply_);

int __xcb_dri2_connect_driver_name_length__(const xcb_dri2_connect_reply_t *_reply_);

xcb_generic_iterator_t __xcb_dri2_connect_driver_name_end__(const xcb_dri2_connect_reply_t *_reply_);

void *__xcb_dri2_connect_alignment_pad__(const xcb_dri2_connect_request_t *_reply_);

int __xcb_dri2_connect_alignment_pad_length__(const xcb_dri2_connect_reply_t *_reply_);

xcb_generic_iterator_t __xcb_dri2_connect_alignment_pad_end__(const xcb_dri2_connect_reply_t *_reply_);

char *__xcb_dri2_connect_device_name__(const xcb_dri2_connect_request_t *_reply_);

int **xcb_dri2_connect_device_name_length**(const xcb_dri2_connect_reply_t *_reply_);

xcb_generic_iterator_t **xcb_dri2_connect_device_name_end**(const xcb_dri2_connect_reply_t *_reply_);

REQUEST ARGUMENTS

conn The XCB connection to X11.

window TODO: NOT YET DOCUMENTED.

driver_type TODO: NOT YET DOCUMENTED.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_DRI2_CONNECT*. 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).

driver_name_length
TODO: NOT YET DOCUMENTED.

device_name_length
TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_dri2_connect_cookie_t*. Errors have to be handled when calling the reply function *xcb_dri2_connect_reply*.

If you want to handle errors in the event loop instead, use *xcb_dri2_connect_unchecked*. See **xcb-requests(3)** for details.

ERRORS

This request does never generate any errors.

SEE ALSO

AUTHOR

Generated from dri2.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.