NAME

xcb_randr_get_screen_resources -

SYNOPSIS

#include <xcb/randr.h>

Request function

xcb_randr_get_screen_resources_cookie_t **xcb_randr_get_screen_resources**(xcb_connection_t *_conn_, xcb_window_t _window_);

Reply datastructure

```
typedef struct xcb_randr_get_screen_resources_reply_t {
    uint8_t        response_type;
    uint8_t        pad0;
    uint16_t       sequence;
    uint32_t       length;
    xcb_timestamp_t timestamp;
    xcb_timestamp_t config_timestamp;
    uint16_t       num_crtcs;
    uint16_t       num_outputs;
    uint16_t       num_modes;
    uint16_t       names_len;
    uint8_t        pad1[8];
} xcb_randr_get_screen_resources_reply_t;
```

Reply function

xcb_randr_get_screen_resources_reply_t *__xcb_randr_get_screen_resources_reply__(xcb_connection_t *_conn_, xcb_randr_get_screen_resources_cookie_t _cookie_, xcb_generic_error_t **_e_);

Reply accessors

xcb_randr_crtc_t *__xcb_randr_get_screen_resources_crtcs__(const xcb_randr_get_screen_resources_request_t *_reply_);

int __xcb_randr_get_screen_resources_crtcs_length__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_generic_iterator_t __xcb_randr_get_screen_resources_crtcs_end__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_randr_output_t *__xcb_randr_get_screen_resources_outputs__(const xcb_randr_get_screen_resources_request_t *_reply_);

int __xcb_randr_get_screen_resources_outputs_length__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_generic_iterator_t __xcb_randr_get_screen_resources_outputs_end__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_randr_mode_info_t *__xcb_randr_get_screen_resources_modes__(const xcb_randr_get_screen_resources_request_t *_reply_);

int __xcb_randr_get_screen_resources_modes_length__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_randr_mode_info_iterator_t __xcb_randr_get_screen_resources_modes_iterator__(const xcb_randr_get_screen_resources_reply_t *_reply_);

uint8_t *__xcb_randr_get_screen_resources_names__(const xcb_randr_get_screen_resources_request_t *_reply_);

int __xcb_randr_get_screen_resources_names_length__(const xcb_randr_get_screen_resources_reply_t *_reply_);

xcb_generic_iterator_t __xcb_randr_get_screen_resources_names_end__(const xcb_randr_get_screen_resources_reply_t *_reply_);

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_RANDR_GET_SCREEN_RESOURCES*. 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).

timestamp TODO: NOT YET DOCUMENTED.

config_timestamp
TODO: NOT YET DOCUMENTED.

num_crtcs TODO: NOT YET DOCUMENTED.

num_outputs TODO: NOT YET DOCUMENTED.

num_modes TODO: NOT YET DOCUMENTED.

names_len TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_randr_get_screen_resources_cookie_t*. Errors have to be handled when calling the reply function *xcb_randr_get_screen_resources_reply*.

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

ERRORS

This request does never generate any errors.

SEE ALSO

AUTHOR

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