Check Whether a Scrollbar Has Reached the Bottom

When working with applications that contain scrollable grids or containers, you may need to scroll through the content until a specific cell becomes visible.

The following function checks whether the specified cell is visible. If the cell is not visible, it scrolls the target vertical scrollbar and continues checking until the cell is reached.

> **Note:** The scrollbar selector used in this example is specific to the application's DOM structure. The index (`[2]`) may need to be adjusted depending on which scrollbar is being targeted.

```
function IsScrollbarAtBottom(VARIABLE) {

    var totalHeight = 1;
    var clientHeight = 0;

    while (totalHeight > clientHeight) {

        var scrollHeight = _eval(
            "document.getElementsByClassName('k-scrollbar k-scrollbar-
vertical')[2].scrollHeight"
        );
        var intScrollHeight = parseInt(scrollHeight);

        var scrollTop = _eval(
            "document.getElementsByClassName('k-scrollbar k-scrollbar-vertical')[2].scrollTop"
        );
        var intScrollTop = parseInt(scrollTop);

        var offsetHeight = _eval(
            "document.getElementsByClassName('k-scrollbar k-scrollbar-
```

```
vertical')[2].clientHeight"
        );
        clientHeight = parseInt(offsetHeight);

        totalHeight = intScrollHeight - intScrollTop;

        log("Scroll height: " + intScrollHeight);
        log("Scroll position: " + intScrollTop);
        log("Client height: " + clientHeight);
        log("Remaining height: " + totalHeight);

        if (isVisible(cell(VARIABLE))) {
            totalHeight = clientHeight;
        } else {
            _eval(
                "document.getElementsByClassName('k-scrollbar k-scrollbar-
vertical')[2].scroll(0,40000)"
            );
        }
    }
}

IsScrollbarAtBottom(VARIABLE);
```

How It Works

The function performs the following operations:

1. Retrieves the total height of the target scrollbar using `scrollHeight`.
2. Retrieves the current vertical scroll position using `scrollTop`.
3. Retrieves the visible height of the scrollbar using `clientHeight`.
4. Calculates the remaining scrollable height.
5. Checks whether the specified cell is visible using `isVisible(cell(VARIABLE))`.
6. If the cell is visible, the function stops.
7. If the cell is not visible, the scrollbar is moved down and the check is repeated.

The following calculation determines how much content remains below the current scroll position:

```
totalHeight = scrollHeight - scrollTop;
```

Identifying the Target Scrollbar

The example uses the following DOM selector:

```
document.getElementsByClassName(
    'k-scrollbar k-scrollbar-vertical'
)[2]
```

The `[2]` selects the third matching scrollbar element on the page. This value is application-specific and may need to be changed if the page contains a different number or order of vertical scrollbars.

Example

To scroll until a specific cell becomes visible:

```
IsScrollbarAtBottom(VARIABLE);
```

Replace `VARIABLE` with the value used to identify the target cell.

> **Important:** Despite the function name, the primary purpose of this implementation is to scroll through a container until the specified cell becomes visible. It should not be considered a generic scrollbar-position API.

Last updated: October 15, 2026

Revision #4
Created 2026-07-31 15:40:00 UTC by Admin
Updated 2026-10-15 16:04:49 UTC by mmora