ItemDetailsBridge Object

ItemDetailsBridge provides api for interaction with the Structure app and item details lifecycle.

Item details lifecycle: enabled (panel was open) visible (ready for interaction) disabled (panel was closed) detached (ItemDetails module deactivated).

ItemDetailsBridge instance is accessible in ItemDetailsProvider subclasses via `'itemDetailsBridge'` property.

Properties

`panel$`

jQuery object that contains item details panel element. All item details viewports are appended to this container.

Methods

`getCurrentRowData()`

Returns `rowData` as returned by corresponding ItemDetailsProvider.extendFocusedRowData() if the current row has the same item type as for which provider is registered, or `null` if it has another type.

`onCurrentRowChange(listener)`

Bind listener to execute each time on changing focused row in Structure.

Parameters

listener	*Function(Object)*	function that accepts `rowData` as returned by ItemDetailsProvider.extendFocusedRowData()

`toggleEnabled(enabled)`

Enable/disable details panel layout.

Parameters

enabled	*Boolean*	`true` if panel should be enabled and `false` otherwise

`onEnableToggling(listener)`

Bind listener to execute each time the panel layout has been enabled or disabled.

Parameters

listener	*Function(Boolean)*	function that accepts `enabled` flag

`isEnabled()`

Returns `true` if panel layout is enabled and `false` otherwise.

`whenVisible(listener)`

Bind listener to execute when details panel becomes ready for interaction.

Parameters

listener	*Function*	function to execute

`isVisible()`

Returns `true` if user can interact with the details panel at the moment and `false` otherwise.

`onDetachment(listener)`

Bind listener to perform cleanup on `ItemDetails` module deactivation.

Parameters

listener	*Function*	function that performs cleanup

notifyDetailsEditStarted()

Should be called when user starts item details content editing. This prevents details refreshing after possible structure updates.

notifyDetailsEditStopped()

Should be called when user has finished item details content editing. This allows the application to perform details refreshes after structure updates. See no
tifyDetailsEditStarted().

notifyDetailsUpdated()

Should be called when user has finished details update. Needed to notify Structure about possible updates to refresh corresponding rows.

notifyFocusChanged(focused)

Should be called when details panel has been focused or lost its focus.

Parameters

enabled	*Boolean*	true if the panel got focus and false otherwise

shouldBlockActions()

Returns true if item details actions should be blocked at the moment.

requestRefreshDetails(deferred)

Request item details refresh. Details panel will be ready for interaction after refresh when corresponding row is updated in Structure. Can be used for Structure and details updates synchronization.

Parameters

deferred	*Object*	jQuery.Deferred that must resolve when details panel has been refreshed and Structure row has been updated

isFocused()

Returns true if details panel is currently focused.

setFocused(focused)

Called to make details panel focused/unfocused.

Parameters

focused	*Boolean*	true if the panel should be focused and false otherwise

focusStealerStarts(stealerId, isFocusedNow)

Details panel focus can be temporarily taken by some UI elements or dialogs and should be returned back after its closing (or it can be some other circumstances). To provide focus retrieval this function should be called when stealer takes focus from details panel. focusStealerFinishes() should be called with the same stealerId to return focus back.

Parameters

stealerId	*String*	identifier of entity that stole focus from details panel
focused	*Boolean*	indicates that details panel should be considered as focused even if it is not so at the moment of function call

focusStealerFinishes(stealerId, preventFocusing)

This function should be called when UI element or dialog that stole focus should return it back (i.e. dialog has been closed or disappeared). See focusSte
alerStarts(). Details panel will be focused back when all stealers release their focus.

Parameters

`stealerId`	*String*	identifier of entity that released focus
`preventFocusing`	*Boolean*	indicates that details panel should not be focused even if all focus stealers were released