ItemDetailsProvider Class

window.almworks.structure.api.ItemDetailsProvider

ItemDetailsProvider class is the extension point for item details.

To use it, create an ItemDetailsProvider subclass via subClass() function, define necessary properties and then register provider subclass for the specific item type using registerItemDetailsProvider().

Example

```
var api = window.almworks.structure.api;
var AttachmentDetails = api.subClass('AttachmentDetails', api.ItemDetailsProvider, {
 init: function () {
   ...
 },
 viewport: viewportElement,
 showDetails: function (rowData, life) {
   ...
 },
 ...
});

api.registerItemDetailsProvider('com.acme.my-plugin:type-attachment', AttachmentDetails);
```

Each ItemDetailsProvider subclass has access to ItemDetailsBridge instance via inherited `'itemDetailsBridge'` property. This object provides additional api for interaction with the Structure app and item details lifecycle.

Properties

`viewport`

Required property. DOM element with the viewport container.

This element should be detached from DOM, Structure itself attaches it to the right place.

`timeoutMessageKey`

Part of i18n key in `'s.itemDetails.stub.title.+$timeoutMessageKey+'` and `'s.itemDetails.stub.body.+$timeoutMessageKey+'` to be used when current row data are loaded for too long.

See extendFocusedRowData(). Has `'timeout'` value by default.

`panelClass`

Class to be set on the outer item details panel div element. Has empty string value by default.

`focusElementClass`

Class of element on the viewport that should be focused when Structure needs to switch focus on the item details panel. Viewport element will be focused if element with this class can't be found or is invisible.

Default value is `'detailsFocusElement'`.

`refreshOnStructureUpdate`

Boolean property. Indicates whether details content must be refreshed via refreshDetails() on Structure update.

Default `false` value means that details will be refreshed only after successful item editing on Structure panel.

Methods

`init()`

Initializer that is called during subclass instance creation.

extendFocusedRowData(rowData)

Returns a jQuery.Deferred that must either: resolve with `rowData`, possibly extended with additional data, or reject with two parameters:

- `'reason'`: *String* - is used as a i18n key in `'s.itemDetails.stub.title.+$reason+'` and `'s.itemDetails.stub.body.+$reason+'`.
- `'isError'`: *Boolean* - if details should display reason message decorated as error (optional, default: `false`).

If the returned promise doesn't resolve or reject in a timely manner, Structure rejects it with argument equal to provider property timeoutMessageKey. If another row is focused before this promise is done, Structure rejects it automatically.

Default implementation returns a resolved deferred object with `rowData`.

Parameters

`rowData`	*Object*	Holds current row data: • `'rowId'`: *Number* • `'itemId'`: *String*

showDetails(rowData, life)

Must be implemented in subclass. The main method - displays the details for the current row in the viewport.

Returns jQuery.Deferred that must resolve when the details are fully shown and the user can interact with them, or reject if the details cannot be shown, with the following parameters:

- `'reason'`: *String* - is used as a i18n key in `'s.itemDetails.stub.title.+$reason+'` and `'s.itemDetails.stub.body.+$reason+'`,
- `'isError'`: *Boolean* - if details should display reason message decorated as error (optional, default: `false`).

If another row is focused while details are being loaded, Structure rejects it.

Parameters

`rowData`	*Object*	Row data as returned by extendFocusedRowData()
`life`	*Object*	Lifespan for showing details - when it finishes, we no longer need to show the details. Function that may cancel details loading or do some cleanup on lifespan finish can be registered via `life.addDetach(function)`

beforeHide(rowData)

Called just before the panel is hidden to perform necessary cleanup.

Parameters

`rowData`	*Object*	Row data as returned by extendFocusedRowData()

hasUnfinishedEdits()

Called before hiding details or switching it to another item to check if details has unfinished edits. Confirm dialog appears before details hiding if method returns `true`.

Should be used to prevent unexpected details hiding during content editing.

getHeaderTitle(rowData)

Returns text (i18nized) to set to the panel title in the details panel header.

If this method is not overridden or returns a falsy value, a title value is set to `'Item Details'`.

Parameters

`rowData`	*Object*	Row data as returned by extendFocusedRowData()

getIcon(rowData)

Returns icon to be displayed in the details header. Return value can be either html string or promise that must be resolved with a html string if it requires asynchronous loading.

The default implementation loads icon html attribute for the focused row.

Parameters

rowData	*Object*	Row data as returned by [extendFocusedRowData()](#)

applyWidth(width)

Updates the displayed details content so it fits the given width. Called on changing details panel width.

Parameters

width	*Number*	New panel width

onDetailsFocusing()

Called after Structure has switched focus to the details panel to perform additional operations. As an option can be used for focusing iframe document if details content is rendered inside it.

refreshDetails(rowData, life)

Performs refresh of details panel content. Called after successful focused item editing on the Structure panel. Called on Structure update only if [refreshOnStructureUpdate](#) is set to `true`.

Returns a [jQuery.Deferred](#) that must resolve when item details has been refreshed.

Parameters

rowData	*Object*	Row data as returned by [extendFocusedRowData()](#)
life	*Object*	Lifespan for showing details - when it finishes, we no longer need to show details. Function that may cancel details loading or do some cleanup on lifespan finish can be registered via `life.addDetach(function)`

setDebug(debug)

Controls additional debug information in logs.

Parameters

debug	*Boolean*	Enable/disable debug logging