Concept - Status bar in subapps

Abstract

This concept is the technical specification for showing a status bar at the bottom of subapp views. General requirements are described as part of the following issue:

> **Error rendering macro 'jira'**
>
> Unable to locate Jira server for this macro. It may be due to Application Link configuration.

(i) **Status**

✅ Ready for implementation

- selection count is either 0 or 1 until proper support for multiple selection
- item count is synchronized by WorkbenchPresenter
    - view.getSelectedView().getContainer().addItemSetChangeListener
    - updates statusBarPresenter
- string template for item/selection count is hard-coded in WorkbenchStatusBarPresenter
    - until proper concept for message keys
- changing color according to table scrolling state requires a client-side patch
    - server does not know when the scrollbar is around
- we don't tackle status bar in page editor for now

Requirements

- In content apps' workbench
    - Show item count, selection count
        - template?
    - Endless sheet of paper UI effect
    - Breadcrumb / path
        - clickable
- In page editor subapp
    - collaboration info (soft-locking)

Proposal

- the `StatusBarView` interface
    - is a UI container that provides three slots for left, center or right-aligned content (raw text or vaadin component)
        - `#addComponent(Component, Alignment)`
            - does not care about vertical alignment
            - first added first aligned
        - `#removeComponent(Component)`
    - has a changeable background color
        - `#setColor(String)`
        - more semantics?
            - FRAME (transparent == grey, default)
            - BLANK (white)
            - INFO (blue)
            - WARNING (yellow)
            - ERROR (red)
        - UI Constants class - actually style name constants
            - ThemeConstants, in vaadin theme
    - is implemented as a vaadin `HorizontalLayout` (i.e. rather than a `CustomComponent`) since the view impl is never exposed directly
    - resides in new *magnolia-ui-statusbar* module

- the `WorkbenchStatusBarPresenter`
  - uses default view implementation to display path (left) and total/selected (right)
    - `#setSelectedItem(Item)`
    - `#setItemCount(int)`
    - `#setSelectionCount(int)`
    - default string template defined in properties file
      - `"{0} item(s), {1} selected"`
      - extendability by content apps
  - view is appended after content views
  - is injected in `WorkbenchPresenter`
    - reacts to selection events as well as workbench resync
    - we try to reuse the eventBus for less tight coupling - we get it from the workbench presenter (can be *chooseDialogEventBus*)
    - might require to fire selected events in a more consistant way, including on workbench resync (e.g. from location)
  - optional: can also fire `ItemSelectedEvent` (breadcrumb)
  - resides in *magnolia-ui-workbench* module

- the `PageEditorStatusBarPresenter`
  - same soft-locking info as in 4.5
    - uses default view
    - can display collaborating users in a "small" dialog or a tooltip
    - #setCollaboratingUsersCount(int)
  - *magnolia-ui-app-pages* or *magnolia-module-softlocking*?
  - requires migration of the soft-locking module

- No configuration

Resources

Status bar can be seen in the following mockups and designs on searching, filtering or pulse.

- Basic search in apps
- Filtering a list of items
- http://wiki.magnolia-cms.com/download/attachments/53258368/Desktop_SG_09_AssetMgmt_MT.png?version=1&modificationDate=1343747055000
- http://wiki.magnolia-cms.com/download/attachments/53258368/Desktop_SG_05_Pulse_MT.png?version=1&modificationDate=1343746824000
- Showing notifications as notes#Thestatechangeindication