

Concept - Documentation of key UI constructs

Patterns

Vaadin

View & Presenter

- Every presenter implements a listener
- A presenter typically has a view as a private member.
- Calling start on a presenter returns the view
- In a presenters start method - it typically sets its views listener to itself. So view's listener is the presenter.

Questions:

What is the idea behind the start method?

View & Vaadin

View is an interface with one method: `asVaadinComponent()`

View is our internal vaadin agnostic class. But every view is backed with a vaadin component.

Definitions

Injection

EventBus - which event bus!

Constructs

There are several areas of the UI that one might get involved in. Code is self documenting - but it helps to have an overview and a reference to the SIGNIFICANT members of each of these constructs.

At least I feel the need in my work as i settle into work on a story - a little preparation in understanding the construct first will be useful. I may as well document it in a way it can be shared, rather than just jotting in my notebook.

Main constructs that could use an overview:

- Shell / `uiContext` / `appController`
- Locations
- App
- Browser
- Workbench
- Actions
- Dialogs & Notifications
- Forms & Fields
- Item / `ItemAdaptor` / etc. / `JcrItem` / `com.vaadin.data.Item`

Workbench

- The workbench is the space to the left of the actionbar in a browser subapp. It typically provides views into the data of a workspace in the repository.
- A standard workbench has 3 contentviews, tree, list, thumbnail + search which show the data and allow selection.
- The data access layer is the Container. Each view gets data from a container implementation - the container talks to the data storage system - currently only JCR is supported.
- `WorkbenchView` extends `VerticalLayout` and assembles the toolbar, the `viewModes`, the searchbox and the statusbar.
- `WorkbenchPresenter` reads its definition and gets all of the `contentPresenters` and their views - orchestrates the assembly.
- A `WorkbenchPresenter` is most often the child of a `BrowserPresenter` or a `WorkbenchChooseDialogPresenter`.

Each contentview has a data and a presenter. We'll consider the tree.

- BrowserSubApp.start
- BrowserPresenter.start
 - WorkbenchPresenter
 - workbenchDefinition
 - view (WorkbenchView)
 - activePresenter (ContentPresenter)
 - contentPresenters (LinkedHashMap<String, ContentPresenter>)
 - TreePresenter (AbstractContentPresenter)
 - getContainer() HierarchicalJcrContainer (AbstractJcrContainer)
 - WorkbenchView
 - contentViews (HashMap<String, ContentView>)
 - ContentView
 - TreeViewImpl (ListViewImpl)
 - TreeTable (vaadin component)
 - dropHandler TreeViewDropHandler

ContentView Interface

- setListener
- select
- setMultiselect
- Listener
 - onItemSelection
 - onDoubleClick
 - onRightClick
 - getIcon