

Concept - Keyboard Shortcuts

Purpose

User can perform most needed actions via convenient keyboard shortcuts.

User can easily learn what the shortcut is for the action they wish to perform.

Standard OS application shortcut keys will be overridden where possible so that M5 behaves like an app as most users will expect.

References & Related

[Initial keyboard support](#)

<https://vaadin.com/book/-/page/advanced.shortcuts.html>

<https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent>

Details

Overridden OS keyboard shortcuts

	Mac	Windows	linux
Save (Dialog)	CMD + S	CTRL + S	
Cancel (Dialog)	CMD + W		
Close Tab	CMD + W		
Close Application	CMD + W		
Copy	CMD + C		
Cut	CMD + X		
Paste	CMD + V		
Ideas			
Next Tab	CTRL + TAB	CTRL + TAB	
Previous Tab	SHIFT + CTRL + TAB	SHIFT + CTRL + TAB	
Specific Tab	CMD + NUMBER	CTRL + NUMBER	

Scoping the Tab "Jump field" key to an area.

- Solution: wrap the entire form in a form element.
 - I think this is best as the tabs are all present in the form the whole time, but only one is visible. When you switch to SHOW ALL - it just shows all of the tab contents.
 - I suggest that when you are at the end of the visible tab of the form and hit tab - it switches you to the field at the top of the next tab.

Questions

- Should these also support TAB key? IE CMD + C, and TAB + C?
- How to handle Tabs in a form? How to handle "heirarchies" of tabs where one set of tabs is in another set of tabs?
Options:
 - Hitting tab keys in a form not only brings you through the fields but also through the tab buttons.
 - Hitting tab keys in a form brings you to next form after the last element.
 - Above specified tab shortcuts operate on the most inner set of tabs.
 - When you get to the end of the tab row - the tab shortcut jumps to the next higher level of tabs.

Shortcuts

ShellApp: Apps	1
ShellApp: Pulse	2
ShellApp: Favorites	3
Meta:	
Next App	Left Arrow
Previous App	Right Arrow
Switch View (In an app)	V to hilight view area, then arrows. (Better because you can add more. Or do they count as tabs?) T, L, I
Jump to list (In an app)	?
Ideas	
to move around on Apps launcher screen.	Arrows + Enter or Space.
Jump to search	S
Fullscreen	F
Preview	ESC
Select All (In list/tree)	A
Help / Info	H? I? I?
Common ActionBar items	
Upload	U
Add (Content node)	A
Add Folder	F
Add Property	P?
Delete	X?
Edit	E
Rename	R
Publish	P
Publish with Subpages	?
Unpublish	U?

Export	Necessary?
Import	Necessary?
Move	M
Duplicate	D?

Keyboard Shortcut Discoverability

Pressing the TAB key makes all keyboard shortcuts appear.

Most shortcuts are activated with TAB + the displayed key.

Must be easy to assign keys for new actions. With Configuration.

Implementation

How does the client side know about all of the available keyboard shortcuts?

Needs this information in order to show the shortcuts on screen in the correct location.

Seems easiest to have some kind of class or something on the element - otherwise how could all of the positioning be done correctly?

Options:

- Each element that can have a keyboard shortcut - implements functionality itself to show the shortcut label. (sounds complicated)
- Each element adds a class - client plugin finds them all and displays the data displayed in the html. (sounds easy)

Keyboard scope must be limited to open shell app or open app.

Questions

How to set different shortcuts depending on OS. META+S on Mac, CTRL+S on Windows, Something on Linux.

How to track TAB key and only handle when no extra keys are pushed? Look at how escape key works now.

SHIFT + TAB? Or other modifiers?

Do we go with always triggering client side buttons so that all actions are guaranteed identical? YES. LESS CODE. KISS.

Problems

No easy way to use TAB as modifier while retaining TAB functionality. Cannot "trigger" a TAB key event.

Must programatically trigger jumping to next field. For this it would be best if all tabbable fields were in forms.

What about when people are in lists/tables. Should also be possible to use TAB here. If its in its own PANEL will that capture the TAB?

Problem - Scope of events

A major problem is the scope of the events, and also getting events not to propogate once they have been handled.

A few explicit examples:

- We had to disable DELTE key shortcut handling, because when you are performing inline editing, then you cannot use the DELETE key anymore to delete characters because the shortcut is active!
 - (One solution would be to add and remove the shortcut handling anytime some starts or ends inplace editing!

- ESCAPE key toggles view and edit mode in page editor. It should also close dialogs. Problem is that event was handled by both the dialog and the page editor! There must be a way to decide the order of who handles the events, and then to stop the propagation of the event - ie stopPropagation() in javascript.
- CTRL-W should close a dialog -then close the active tab - then close the app. And maybe finally if you are in app launcher - it should present a javascript confirmation if you really want to close the tab to prevent people from accidentally closing the tab.

Organization of Work

A list of the tasks

- Come up with a concept / pattern for how to implement the keyboard shortcuts
 - serverside
 - clientside
 - clientside - but configured from server.
- Things that should be included:
 - ESCAPE should toggle between preview & edit mode. SERVER-SIDE.
 - Forms and dialog should react to CTRL+S
 - Browser Tab / App should react to CTRL+W
 - ENTER while on element should trigger default option.
 - TAB while on tree field should step to next element.
 - Dialog should close with ESCAPE - Verify.
- Given that there is not much time - we should go ahead and implement what we have using the established server pattern. The Tab key shortcut magic will come later - hopefully.

Decisions guiding where shortcuts are implemented

How does one capture CMD and CTRL shortcuts? This does work in Vaadin (and probably HTML) META key is the osx cmd key.

How would the TAB implementation work:

A big challenge is scoping the TAB - both showing the tips, and reacting to simple TAB presses.

- Serverside - Using Vaadin:
 - I see no way
- GWT with widgets:
 - I dont see a handy way to scan though all of the controls to show their tooltips.
 - GWTQuery or JQuery. Then every item just has to have an accesskey applied.
- Do we have to have client side and server side handling? How could client side handle the CTRL / CMD overriding as well.

How to make keyboard shortcuts configurable. (Do they need to be?) (on a user level?) Per language would be good. Or simply ability to configure through some kind of property file.

Challenges

Scoping the Tab "Jump field" key to an area.

- Solution: wrap it in a form element.
 - But then how to handle the tabs of the magnolia form?
 - Should the form include the tabs? No - each tab should have its own form - we only want to step through the fields of the one tab usually.
 - How to move between the form tabs? - Hitting tab when you are at the end of the form could cycle through the form tabs.

Scoping the Tab Shortcut field to the visible controls. Shell + current App + current SubApp.

Should CMD handling also happen on client so that we dont have two keyboard shortcut systems?

How would CMD+S work on browser? How do you scope it.

- For example: How to handle
 - a dialog open in two different apps. Which dialog do you save on CTRL-S?
 - An app, a subapp, a dialog open. on CTRL-W, which one do you close?
- On a browser you could scope the events in two ways:
 - Add/Remove
 - Add the event listener when the app/subapp/dialog is focussed.

- Remove it when it loses focus - ie another app comes to the foreground.
- Check for top item:
 - In the handler - check which item is "on top", is focussed

Notes

- InplaceEditingTreeTable - where ESC, ENTER, TAB and SHIFT-TAB are handled.
- WorkbenchViewImpl - where keyboardEventPanel lives.
 - handlers for ENTER and DELETE key.
 - ListViewImpl - Handles Enter key with onShortcutKey() - passes to presenter.
 - AbstractContentPresenter.onShortcutKey().
 - Fires ItemShortcutKeyEvent() on the eventbus!
 - BrowserPresenter adds a handler but simply fires executeDefault action - should check the event for ENTER key.
 - Now i just added an executeDeleteAction() method.