Ausarbeitung im Rahmen des Studienfachs Web-Techniken

Thema: Konkrete Umsetzung einer Seitennavigation mit Vaadin-Touchkit

Dozent: Stephan Schiffner Student: Sebastian Stumpf

Projektgruppe: Vaadin Touchkit

Ich habe mich für die Thematik entschieden, da sich in Vaadin Touchkit die zu Erstellung einer Navigation verfügbaren Mechanismen nicht auf den ersten Blick erschließen. In unserem Projekt, der Umsetzung eines "Domination" Spieles als an Mobile Devices angepasste Webanwendung ist das nicht ins Gewicht gefallen. Wir haben unsere eigene Navigation gebaut, welche besser auf die Bedürfnisse des Projektes angepasst ist.

Die verwendeten Klassen/Komponenten:

Alle hier verwendeten Komponenten sind Vaadin Touchkit Components und deren Style somit speziell auf mobile Endgeräte angepasst.

TabBarView:

Die TabBarView ist eine Klasse in Vaadin Touchkit, die eine Tab Navigation am unteren Bildschirmrand anbietet. Den Tabs können Namen und Icons zugewiesen werden.

Beim Tabwechsel wird der Zustand des verlassenen Tabs gespeichert, so dass dieser beim Wechsel zurück wieder geladen wird. Dies gilt bei der Verwendung des NavigationManagers auch für die aktuelle Navigationsebene.

NavigationManager:

Der NavigationManager ist die Basiskomponente für die Navigation zwischen verschiedenen NavigationViews in Vaadin Touchkit, welche sich immer die letzte Seite merkt und automatisch einen "Zurück"-Button zu dieser erstellt. Eine Instanz des aktuellen Navigationmanagers bekommt man bei der Zuweisung des Listeners an einen NavigationButton, bei der Implementierung des entsprechenden Events (z.B. Click-Event) desselben, durch Aufruf von getNavigationManager().

Durch einen Fehler in Vaadin Touchkit liefert getNavigationManager() in der NavigationView leider nur eine Null-Referenz, weshalb man obigen Weg wählen sollte.

Der NavigationManager speichert sich 3 Views, previous- current- und nextView, welche bei einem Navigationsevent automatisch angepasst werden.

Auf NavigationEvents kann durch binden eines NavigationEventListeners an den NavigationManager auch nach eigenen Bedürfnissen auf die Events reagiert werden.

Im Beispiel wird z.B. die Anzahl der Vorwärts- und Zurück-Navigation Events gezählt.

NavigationView:

Die NavigationView bietet in der Headerleiste Platz für die Überschrift und zwei Komponenten, RightComponent undLeftComponent. LeftComponent wird bei der Verwendung eines NavigationManagers bei einem NavigationEvent vorwärts automatisch mit einem Zurück-Button befüllt.

In der NavigationView kann die RightComponent sowie ein NavigationButton im Content für die Navigation verwendet werden.

NavigationButton:

Das Standardelement für die Navigation im Content einer Navigationview. Um den gelungenen Style zu verwenden, sollte er in einer von Vaadin bereit gestellten VerticalComponentView eingebunden werden.

Beim binden eines ClickEventListeners kann auf den NavigationManager zugegriffen werden.

Die Implementierung:

Initialisierung der Tabs in der TabBarView:

```
** Die Tabs im Konstruktor der von TabBarView erbenden Klasse setzen
*/
public MainTabSheet() {

    // Die currentView des NavigationManagers wird auf die im Konstruktor
    // übergebene Navigation View gesetzt
    Tab addTab = addTab(new NavigationManager(new NavigationView1()));
    addTab.setCaption("Tab 1");
    // Den Inhalt des initialen Tabs festlegen
    setSelectedTab(addTab);

    addTab = addTab(new NavigationManager(new NavigationView2()));
    addTab.setCaption("Tab 2");
    ...
}
```

Erstellen eines NavigationButtons im Content einer NavigationView:

```
NavigationButton myButton = new NavigationButton("Navigate to ... ");
myButton.addListener(new Button.ClickListener() {
    @Override
    public void buttonClick(final Button.ClickEvent event) {
        getNavigationManager().navigateTo(new NavigationView11());
    }
});
content.addComponent(myButton);
```

Erstellen eines NavigationButtons als rechte Komponente der NavigationView:

```
Button myButton = new Button("-> to 111 Level 3 (11)");
myButton.addListener(new Button.ClickListener() {
     @Override
     public void buttonClick(final Button.ClickEvent event) {
         getNavigationManager().navigateTo(new NavigationView111());
     }
});
setRightComponent(myButton);
```

Hier ist darauf zu achten die Klasse Button zu verwenden und nicht NavigationButton, da der Style des NavigationButtons nicht an die Darstellung als Right Component der NavigationView geeignet ist.

Reagieren auf Navigation Events im NavigationManager:

Mit den vorgestellen Komponenten und Konstrukten können nun recht einfach Webanwendungen geschrieben werden, in denen die Inhalte über mehrere Ebenen durch eine Navigation verbunden sind.

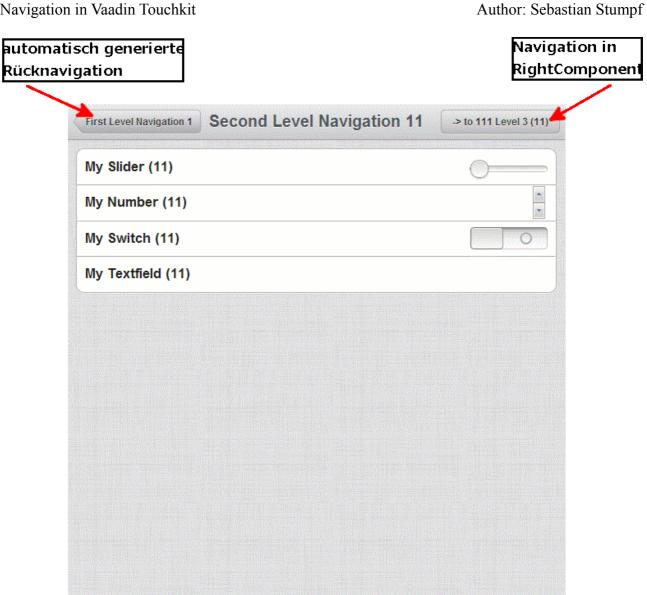
Die Umsetzung:

Das Ergebnis einer beispielhaften Implementierung in Vaadin Touchkit, welche eine TabBarView für die erste Navigationsebene und NavigationButtons für folgende Ebenen verwendet wird hier kurz vorgestellt.



Über die TabNavigation erreicht man die Übersichtsseiten der Applikation, in denen man sich über NavigationButtons tiefer hinein navigieren kann.

Beim Wechseln zwischen den Tabs wird die Navigationstiefe des aktuellen Tabs gespeichert. Wenn man wieder zurücknavigiert wird diese also wieder angezeigt wie beim verlassen des Tabs.



Navigiert man über den NavigationButton eine Ebene tiefer wird automatisch ein "Zurück"-Button erstellt, über den man auf die letzte Seite kommt. Diese Funktionalität kommt über den eingebundenen NavigationManager in die Applikation.

Second Level Navigation 11

Third Level Navigation 111

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Summae mihi videtur inscitiae. Non potes, nisi retexueris illa. **Si quidem, inquit, tollerem, sed relinquo.** Nam Pyrrho, Aristo, Erillus iam diu abiecti. <u>Duo Reges: constructio interrete.</u> Honesta oratio, Socratica, Platonis etiam. Primum divisit ineleganter; <u>Sed tamen intellego quid velit.</u>

Idem adhuc:

Non enim iam stirpis bonum quaeret, sed animalis.

Quid iudicant sensus?

Praeclare Laelius, et recte sofñw, illudque vere: O Publi, o gurges, Galloni! es homo miser, inquit.

Confecta res esset.

Deque his rebus satis multa in nostris de re publica libris sunt dicta a Laelio.

ALIO MODO.

Quae hic rei publicae vulnera inponebat, eadem ille sanabat.

Quid Zeno?

Dici enim nihil potest verius.

Zenonis est, inquam, hoc Stoici. <u>Avaritiamne minuis?</u> Uterque enim summo bono fruitur, id est voluptate. Istam voluptatem perpetuam quis potest praestare sapienti? <u>Quibusnam praeteritis?</u> An eiusdem modi? Earum etiam rerum, quas terra gignit, educatio quaedam et perfectio est non dissimilis animantium.

- Quid enim de amicitia statueris utilitatis causa expetenda vides.
- 2. Quasi vero, inquit, perpetua oratio rhetorum solum, non etiam philosophorum sit.
- 3. In eo enim positum est id, quod dicimus esse expetendum.

Comprehensum, quod cognitum non habet? Memini me adesse P. Quamquam ab iis philosophiam et omnes ingenuas disciplinas habemus; Itaque contra est, ac dicitis; Sit enim idem caecus, debilis. Conferam tecum, quam cuique verso rem subicias; Sequitur disserendi ratio cognitioque naturae; Est enim tanti philosophi tamque nobilis audacter sua decreta defendere. Nos vero sum possius esse utilitatem in historia, pon mode voluntatem.

Tab 1 Tab 2 Tab 3

Der Vollständigkeit halber hier noch eine View aus der dritten Navigationsebene, die über den RightComponent Button erreicht wurde.

Ouellen:

Vaadin Touchkit Demo:

http://demo.vaadin.com/vornitologist/VAADIN/tutorial/touchkit-tutorial.html

Vaadin Touchkit Referenz im Book of Vaadin:

https://vaadin.com/book/-/page/mobile.html

Implementierung liegt in folgendem git Repository:

https://github.com/BastusIII/NavigationExample