Tag-Visualisierung und –Befüllung

V = View

V-CB = View-Code-Behind

VM = ViewModel

**Welche Menüaktionen sind in welchen Ebenen möglich**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer | Artist | Title | Genre | Attribute |
| 0 | Type | Type | Type | Type |
| 1 | InputField | InputField | TopGenres | Attributes |
| 2 | Suggestions | Suggestions | SubGenres | Input or select |
| 3 | Edit (Keyword assigned) | Edit (Keyword assigned) | Edit (Keyword assigned) | Edit (Keyword assigned) |
|  |  |  |  |  |

**ToDo**

* Dave fragen:
  + Methode, mit der ich für ein Genre alle Subgenres bekomm
* SearchOptionWorker: Iterierung durch Auswahloptionen, Update der TagOptions etc.

**Wie kann Datenbindung bei den Tags umgesetzt werden?**

* + Variante 1: Drei Views
    1. SearchTagView: TagVisualization als V
       - In TagService laden
       - V-Code-Behind wird geladen
    2. SearchTagsView: V + VM <- In diesen SearchTagViews laden
    3. SearchView: V + VM
  + Variante 2: Zwei Views
    1. 1. Lösung: Binding für jede Variable kann dynamisch gesetzt werden. Es kann hier an verschiedene Elemente einer Liste/ObservableCollection des VM gebunden werden
       - Problem: PieMenuItem Header/SubHeader + Id binden
    2. SearchTagView: TagVisualization + V + VM
       - V-Code-Behind wird jedes Mal bei Tag-Setzung geladen (auch bei einem Tag nach Entfernung mehrmals)
         * Jeder Tag hat seine eigene Instanz des V-CB – nötig?
         * Kein eigenes VM geladen
         * Nutzt SearchVM
       - OnVisualizationLoaded
         * Setze Werte
    3. SearchView: V + VM
  + Variante 3:
    1. SearchTagView: TagVisualization + V
    2. SearchView: V + VM
       - Tangible-Werte werden im V + VM gesetzt. Hier besteht Zugriff auf alle TagVisualizations
* Manuell dynamisch setzen (einfachste Lösung)
  + Im VM Variablen setzen (ObservableCollection Tags)
    1. Im V VM auslesen und V-Elemente setzen (new Binding.. pieMenuItem.Name=binding..)
    2. In XAML-View Variablen als Dynamic Resources verwenden