

DocumentationTree - DV-Konzept

Jens Kötterheinrich

2018-02-20

Inhalt

1. Allgemeines	1
1.1. Zweck und Struktur des Dokuments	1
2. Übersicht	1
3. Funktionalität	1
4. Dateibeschreibungen	1
4.1. LuaTable	1
5. Programmdesign	2
5.1. Ablauf	2
5.2. Benutzeroberfläche	2

1. Allgemeines

1.1. Zweck und Struktur des Dokuments

Dieses Dokument beschreibt die Anwendung DocumentationTree. Die Anwendung wird als Bestandteil von Dokumentation von Datenbanken verwendet. Mit diesem Dokument soll der Leser in die Lage versetzt werden, - den Aufbau der Anwendung zu verstehen, - die Aufgaben der Anwendung, - das System über die Benutzeroberfläche bedienen zu können, - die Schnittstellen zu anderen Systemen zu kennen, - im Falle eines Systemfehlers die geeigneten Maßnahmen ergreifen zu können.

2. Übersicht

Die Anwendung DocumentationTree basiert auf der Java Standard Edition. Es wird mindestens die Version 7 vorausgesetzt. Bei der Anwendung handelt es sich um eine JavaFX-Anwendung. Sie lässt sich mit einer Java Runtime (\geq Version 7) starten.

3. Funktionalität

Die Anwendung stellt folgende Funktionalitäten bereit:

- Öffnen und Parsen einer Datei mit LuaTable (später auch XML)
- Darstellung als Baumstruktur
- Markieren von Knoten in unterschiedlichen Farben zur schnellen Übersicht und Analyse

4. Dateibeschreibungen

4.1. LuaTable

Eine Lua-Datei besteht aus Lua-Code. Darin wird eine LuaTable erzeugt und zurückgegeben. Der Aufbau der LuaTable basiert auf [IupTree](#).

Beispiel LuaTable

```
example =
{branchname="example",
  {branchname="foo", 'Hello'},
  {branchname="bar", 'world!'}
}

return example
```

5. Programmdesign

5.1. Ablauf

5.1.1. Auswahl einer Datei

Nach dem Start der Anwendung wird ein Dialog angezeigt, um eine Datei auszuwählen. Die erlaubten Endungen der zu öffnenden Dateien sind *.lua* und *.xml*.

5.1.2. Parsen einer ausgewählten Datei

Mit Hilfe des Frameworks

```
<dependency>
  <groupId>org.luaj</groupId>
  <artifactId>luaj-jse</artifactId>
  <version>3.0.1</version>
</dependency>
```

wird die LuaTable geparkt und eine Instanz von *org.luaj.vm2.LuaTable* erzeugt.

5.1.3. Erzeugen eines Baums

Auf Basis der erzeugten Instanz der LuaTable werden *TreeItems* erzeugt. Dabei werden die Elemente der LuaTable rekursiv ausgewertet. Einzelne Knoten werden als *TreeItems* erzeugt, die zu einer *TreeView* hinzugefügt werden. Standardmäßig werden Knoten immer aufgeklappt dargestellt. Durch *state = 'COLLAPSED'* kann für einen Knoten definiert werden, dass er initial nicht ausgeklappt dargestellt wird.

5.1.4. Darstellung

5.1.5. Knoten selektieren

5.1.6. Knoten markieren

5.1.7. Fehlerbehandlung

5.2. Benutzeroberfläche