

Entwurfsbeschreibung

Swp 12-2 ColaNut

1. Allgemeines

Die zu entwickelnde Web-GUI stellt eine Bedienungs Oberfläche für das vorhandene und auf Java basierende Programm Limes dar. Sie läuft Serverseitig, sodass nur minimale Datenmengen zum Client gesendet werden müssen. Als Clients kommen, im Allgemeinen, alle gängigen Browser in Frage. Die Web-GUI sieht eine Nutzerverwaltung und eine Datenhaltung von individuellen Nutzerdaten vor, welche in zukünftigen Releases ausgebaut und Funktionstüchtig eingefügt wird. Weiterhin werden alle derzeitigen Funktionen von Limes unterstützt und können problemlos aufgerufen und ausgeführt werden.

2. Produktübersicht

Die Web-GUI wird für eine vereinfachte Nutzung von Limes erstellt und soll dieses Programm zugänglicher und intuitiver Gestalten. So werden verschiedene Tabs zur Gruppierung von Limes internen Funktionen genutzt. Weiterhin können verschiedene Methoden zur Bearbeitung der „LinkSpec“ - Eigenschaften per Drag and Drop zusammengefügt werden. Ein Log-In Screen wurde zum Zugriff auf den Softwareprototyp ebenfalls eingefügt, welcher die Grundlage für die Erweiterung zur Nutzerverwaltung bildet. Das vorhandene Design wird voraussichtlich in zukünftigen Releases im Grundlegenden beibehalten, mit Ausnahme kleinerer, während des Arbeitsverlaufs auffallender Verbesserungen der Ansicht, sowie Verbesserungen der Nutzbarkeit.

3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem / der einzelnen Pakete

Die Web-GUI wird mit dem Play! Webframework erstellt. Dieses Webframework basiert auf Java und HTML, besitzt allerdings eine Framework eigene „vereinfachte“ Syntax. Dieses Framework basiert auf dem Model-View-Controller Designpattern. Es gibt eine große Auswahl an Plug-Ins und Erweiterung für dieses Framework sowie eine einfachere Nutzung verschiedener Datenbanken. Die angestrebte

Datenbank für die Web-GUI wird Apache-Derby werden. Ein wichtiger Aspekt, ist die, dem Play!-Framework eigene, Syntax, welche eine vereinfachte Programmierung der GUI zulässt. Der spezielle Aufbau von Play! Anwendung muss auch beachtet werden, da einzelne Templates, welche genutzt werden oder eingebundene Java-Pakete in Bestimmten Ordnern im Play! Verzeichnis zu finden sind. Das heißt also, dass Play! eine eigene Ordnerstruktur besitzt, welche folgend aussieht.

```

app
  ↳ assets
    ↳ stylesheets
    ↳ javascripts
  ↳ controllers
  ↳ models
  ↳ views
conf
  ↳ application.conf
  ↳ routes
public
  ↳ stylesheets
  ↳ javascripts
  ↳ images
project
  ↳ build.properties
  ↳ Build.scala
  ↳ plugins.sbt
lib
logs
  ↳ application.log
target
  ↳ scala-2.9.1
    ↳ cache
    ↳ classes
    ↳ classes_managed
    ↳ resource_managed
    ↳ src_managed
test

```

- Application sources
- Compiled asset sources
- Typically LESS CSS sources
- Typically CoffeeScript sources
- Application controllers
- Application business layer
- Templates
- Configurations files
- Main configuration file
- Routes definition
- Public assets
- CSS files
- Javascript files
- Image files
- sbt configuration files
- Marker for sbt project
- Application build script
- sbt plugins
- Unmanaged libraries dependencies
- Standard logs folder
- Default log file
- Generated stuff
- Compiled class files
- Managed class files (templates, ...)
- Managed resources (less, ...)
- Generated sources (templates, ...)
- source folder for unit or functional tests

Vertiefend ist weiterhin zu erwähnen, dass zur allgemeinen Gestaltung der GUI ebenfalls jQuery UI 1.8.13 genutzt wurde, einem Toolkit, welches auf der JavaScript jQuery – Bibliothek aufgesetzt wurde und für low-level Interaktion und Animation genutzt werden kann. Dies wurde auch für die Erstellung der Tabs, welche zur Gruppierung genutzt werden und für die eingebetteten Buttons verwendet. Für die Darstellung und Funktion der Metric fand jsPlumb, ebenfalls auf der jQuery – Bibliothek basierend, Einsatz, welches eine Drag and Drop – Funktion mit sich bringt. Zur Kommunikation der GUI mit Java Applikationen (in diesem Fall Limes) wurde JavaScript Object Notation (JSON), genauer JSON2 genutzt und dazu GSON, da zum einen GSON eine Umwandlung von JSON – Objekten in Strings und umgekehrt ermöglicht

und JSON ein einfach zu lesendes, schreibendes und parsendes Datenaustauschformat. Die Grundlegende Basis bildet, im Allgemeinen, die bisherige Version von ColaNut, welche komplett auf JavaScript basiert. Ansonsten sind noch die Limes-Bibliotheken enthalten, welche für die Funktionalität von Limes unerlässlich sind.