



University of Applied Sciences

SEAR^CH–Tag EEXCESS SECH–Browser

Programmier Leitfaden Spezifikation, Konstruktion

Prof. Dr. Peter Stöhr Gottfried von Recum
Alexander Pöhlmann Lothar Mödl Burak Erol
Brian Mairhörmann Andreas Netsch Philipp Winterholler
 Andreas Ziemer

Version 0.4.1

Inhaltsverzeichnis

I. Einleitung	2
1. Übersicht	3
1.1. Abschnitt	3
1.1.1. Teilabschnitt	3
1.1.1.1. Unterteilabschnitt	3
1.1.1.1.1. Paragraph	3
1.1.1.1.1.1. Unterparagraph	3
2. User Interface	4
2.1. Abschnitt	4
2.1.1. Teilabschnitt	4
2.1.1.1. Unterteilabschnitt	4
2.1.1.1.1. Paragraph	4
2.1.1.1.1.1. Unterparagraph	4
3. Programmlogik	5
3.1. Abschnitt	5
3.1.1. Teilabschnitt	5
3.1.1.1. Unterteilabschnitt	5
3.1.1.1.1. Paragraph	5
3.1.1.1.1.1. Unterparagraph	5
II. Spezifikation	6
4. Übersicht	7
4.1. Abschnitt	7
4.1.1. Teilabschnitt	7
4.1.1.1. Unterteilabschnitt	7
4.1.1.1.1. Paragraph	7
4.1.1.1.1.1. Unterparagraph	7

5. User Interface	8
5.1. Abschnitt	8
5.1.1. Teilabschnitt	8
5.1.1.1. Unterteilabschnitt	8
5.1.1.1.1. Paragraph	8
5.1.1.1.1.1. Unterparagraph	8
6. Programmlogik	9
6.1. Abschnitt	9
6.1.1. Teilabschnitt	9
6.1.1.1. Unterteilabschnitt	9
6.1.1.1.1. Paragraph	9
6.1.1.1.1.1. Unterparagraph	9
III. Konstruktion	10
7. Übersicht	11
8. User Interface	12
8.1. ViewController	12
8.1.1. SearchTableViewController	12
8.1.2. ViewController	12
8.1.3. PopViewController	12
8.1.4. SettingsController	12
8.1.5. main.js	12
8.1.6. readHead.js	12
8.2. Delegate	13
8.2.1. AppDelegate	13
8.2.2. WebViewDelegate	13
8.2.3. FavTableDelegate	13
8.2.4. SechTableDelegate	13
8.3. DataSource	14
8.3.1. FavTableDataSource	14
8.3.2. SechTableDataSource	14
8.4. Components	15
8.4.1. FavoriteCell	15
8.4.2. AdressBar	15
8.4.3. SearchCell	15
8.5. Persistence	16
8.5.1. FavoritesModel	16
8.6. WebContent	17

9. Programmlogik	18
9.1. TaskCtrl	18
9.2. SEARCHExtraction	19
9.2.1. SearchManager	19
9.2.2. RegexForSEARCH	19
9.3. SEARCHModels	20
9.4. QueryBuild	21
9.4.1. QueryBuildCtrl	21
9.5. SearchQuerys	22
9.6. QueryResolution	23
9.6.1. JSONData	23
9.6.2. ResponseParse	23
9.6.2.1. FarooResponseParser	23
9.6.2.2. DuckDuckGoResponseParser	23
9.6.2.3. EexcessResponseParser	23
9.6.2.4. AbstractResponseParser	23
9.6.3. QuerySend	23
9.6.3.1. AbstractConnectionCtrl	23
9.6.3.2. JsonConnectionCtrl	23
9.6.3.3. URLConnectionCtrl	23
9.6.4. QueryProcessing	23
9.6.4.1. AbstractBuilder	23
9.6.4.2. AbstractJSONBuilder	23
9.6.4.3. AbstractURLBuilder	23
9.6.4.4. EexcessJSONBuilder	23
9.6.4.5. FarooURLBuilder	23
9.6.4.6. DuckDuckGoURLBuilder	23
9.6.4.7. EEXCESSOrigin	23
9.7. SearchResults	24
9.8. Ranking	25
9.8.1. Rules	25
9.8.2. SearchRules	25
9.8.3. RankingPersistence	25
9.8.3.1. PersistenceController	25
9.8.3.2. RankingDataObject	25
9.8.3.3. RankingDataObjectPersistency	25
9.9. SettingsModel	26

Teil I.

Einleitung

1. Übersicht

1.1. Abschnitt

1.1.1. Teilabschnitt

1.1.1.1. Unterteilabschnitt

1.1.1.1.1. Paragraph

1.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

2. User Interface

2.1. Abschnitt

2.1.1. Teilabschnitt

2.1.1.1. Unterteilabschnitt

2.1.1.1.1. Paragraph

2.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

3. Programmlogik

3.1. Abschnitt

3.1.1. Teilabschnitt

3.1.1.1. Unterteilabschnitt

3.1.1.1.1. Paragraph

3.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

Teil II.

Spezifikation

4. Übersicht

4.1. Abschnitt

4.1.1. Teilabschnitt

4.1.1.1. Unterteilabschnitt

4.1.1.1.1. Paragraph

4.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

5. User Interface

5.1. Abschnitt

5.1.1. Teilabschnitt

5.1.1.1. Unterteilabschnitt

5.1.1.1.1. Paragraph

5.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

6. Programmlogik

6.1. Abschnitt

6.1.1. Teilabschnitt

6.1.1.1. Unterteilabschnitt

6.1.1.1.1. Paragraph

6.1.1.1.1.1. Unterparagraph

1. Eine
2. kleine
3. Aufzählung

Teil III.

Konstruktion

7. Übersicht

Die Konstruktion gliedert sich in drei Abschnitte:

1. Übersicht
2. User Interface
3. Programmlogik

8. User Interface

8.1. ViewController

8.1.1. SearchTableViewController

8.1.2. ViewController

8.1.3. PopViewController

8.1.4. SettingsController

8.1.5. main.js

8.1.6. readHead.js

8.2. Delegate

8.2.1. AppDelegate

8.2.2. WebViewDelegate

8.2.3. FavTableDelegate

8.2.4. SechTableDelegate

8.3. DataSource

8.3.1. FavTableDataSource

8.3.2. SechTableDataSource

8.4. Components

8.4.1. FavoriteCell

8.4.2. AdressBar

8.4.3. SearchCell

8.5. Persistence

8.5.1. FavoritesModel

8.6. WebContent

9. Programmlogik

9.1. TaskCtrl

9.2. SEARCHExtraction

9.2.1. SearchManager

9.2.2. RegexForSEARCH

9.3. SEARCHModels

9.4. QueryBuild

9.4.1. QueryBuildCtrl

9.5. SearchQueryys

9.6. QueryResolution

9.6.1. JSONData

9.6.2. ResponseParse

9.6.2.1. FarooResponseParser

9.6.2.2. DuckDuckGoResponseParser

9.6.2.3. EexcessResponseParser

9.6.2.4. AbstractResponseParser

9.6.3. QuerySend

9.6.3.1. AbstractConnectionCtrl

9.6.3.2. JsonConnectionCtrl

9.6.3.3. URLConnectionCtrl

9.6.4. QueryProcessing

9.6.4.1. AbstractBuilder

9.6.4.2. AbstractJSONBuilder

9.6.4.3. AbstractURLBuilder

9.6.4.4. EexcessJSONBuilder

9.6.4.5. FarooURLBuilder

9.6.4.6. DuckDuckGoURLBuilder

9.6.4.7. EEXCESSOrigin

9.7. SearchResults

9.8. Ranking

9.8.1. Rules

9.8.2. SearchRules

9.8.3. RankingPersistence

9.8.3.1. PersistenceController

9.8.3.2. RankingDataObject

9.8.3.3. RankingDataObjectPersistency

Das Ranking dient der Sortierung der Suchergebnisse nach den Interessen des Nutzers. Das Sortieren der Zusatzinformationen wird durchgeführt, bevor die Ergebnisse dem Nutzer präsentiert werden. D.h. es wird in der Klasse „SearchResults“ initialisiert und gestartet. Die Funktionalität des Rankings wird mit Hilfe von verschiedenen Regeln realisiert. Aktuell existieren drei Regeln: „Language“, „Mendeley“ und „MediaType“. Jede Regel besitzt einen Faktor für die Gewichtung. Dieser legt fest, wie wichtig eine Regel ist. Für jeden Datensatz werden die o.g. Regeln mit den entsprechenden Eingangsparametern erzeugt. Im weiteren Fortschritt des Projekts soll das Ranking dahingehend erweitert werden, dass nicht für jeden Datensatz alle Regeln erzeugt werden, sondern nur diejenigen, die für den jeweiligen Datensatz notwendig sind. So wird z.B. für einen Datensatz, der nicht von der Suchmaschine Mendeley kommt, auch nicht die Regel „Mendeley“ erzeugt. Die erstellten Regeln pro Datensatz werden in einem Hilfs-Array zwischengespeichert (SearchRules). Der Hauptteil des Rankings beginnt erst mit der Berechnung. Trifft eine Regel zu, d.h. der erwartete Wert der jeweiligen Regel entspricht dem tatsächlichen Wert des Suchergebnisses, so wird eine 1 von der Regel zurückgeliefert. Trifft eine Regel nicht zu, so wird eine 0 zurückgeliefert. Dieser Wert wird mit der Gewichtung der jeweiligen Regel multipliziert. Im Anschluss werden alle Ergebnisse aller Regeln, die zu einem Datensatz gehören aufsummiert und durch die Anzahl der verwendeten Regeln dividiert. Dieser errechnete Wert gibt prozentual wider, in wie weit das Suchergebnis den Nutzer interessiert. Nach dem Wert werden alle Suchergebnisse zugehörig zu einem Search-Link der Größe nach sortiert und im Anschluss dem Nutzer präsentiert. Die Klassen "PersistenceController", "RankingDataObject" und "RankingDataObjectPersistency" werden bei dem aktuellen Stand des Projektes noch nicht verwendet. Im weiteren Verlauf werden die Klassen zum Speichern von nutzerspezifischen Informationen verwendet.

9.9. SettingsModel

Abbildungsverzeichnis

Tabellenverzeichnis