

Change Log

Gennadi Heimann

29.12.2017

1 v0.0.1

@created on 14.11.2017

@finished on

1.1

@created on 14.11.2017

@finished on 21.12.2017

Der nächste Schritt wird erst freigegeben, wenn alle Abhängigkeiten der Komponenten geprüft sind. Beim Prüfen von der ausgewählte Komponente wird auch die SelectionCriterium geprüft, wieviel Komponenten in dem Schritt ausgewählt werden können.

Die ausgewählte Komponente wird in der CurrentConfig gesichert, damit das SelectionCriterium bewertet kann.

2 v0.0.2

2.1 Requirements

2.1.1

@created on 4.12.2017

@finished on

Die aktuelle Konfiguration behält gesamte Information über die hinzugefügte Komponente. Dazu gehören alle Abhängigkeiten und Einschränkungen und Komponente mit *< exclude >* Markierung.

2.1.2

@created on 13.12.2017

@finished on

Jede Schritt in dem Konfigurator kann eine Abhängigkeit zu dem Schritt oder Komponente haben. Das bedeutet, dass geladene Schritt eine Komponente oder einen Schritt ausschließen oder fordern.

2.1.3

@created on 18.12.2017

@finished on

Bei dem Abschluss der Konfiguration muss auf die Konsistenz den letzten Schritte geachtet werden. Es kann sein, dass in einem Schritt paar Komponente ohne weiteren Schritt und paar mit weiterem Schritt vorhanden. (siehe szenario_3)

2.1.4

@created on 20.12.2017

@finished on

Bei der Auswahl der gleichen Komponente innerhalb eines Schrittes muss erkannt werden und eine Info an den Webclient gesendet.

Bei der *< Dependency >* werden die Parameter *< message >* und *< status >* nicht gebraucht. Sie können auch gelöscht werden.

2.1.5

@created on 21.12.2017

@finished on 21.12.2017

Zu der *< ComponentOut >* die *< componentId >* und *< stepId >* Parameter bei allen Status hinzufügen.

2.1.6

@created on 21.12.2017

@finished on

Abwählen der Komponente muss auch implimentiert werden. Die Komponente wird auch aus der *< CurrentConfig >* entfernt.

2.1.7

@created on 21.12.2017

@finished on

Bei der Aufruf von *< NextStepIn >* der Web-Client braucht die ID vom vorherigen Schritt. Damit diese beim Bedarf vor dem neuen Schritt gelöscht werden kann.

2.1.8

@created on 21.12.2017

@finished on

Einführung von ODBOName. Es bei jeder Aktion nur einmal die Daten aus der DB geholt und in ein ODBOName Objekt gemapt. Die Aktion arbeitet nur mit diesem Objekt ohne zusätzlich in der DB abzufragen.

2.2 Test

2.2.1 Szenario v002-1

Diese Szenario nur möglich bei der Einstellung von `< configurationCourse = sequence >`.

Es gibt folgende Verläufe: `< sequence >`, `< substitute >`

1. Server: Start Konfiguration – > Schritt 1
2. Client: Auswahl – > Komponent 1-1
3. Server: Komponent 1-1 hinzugefügt, aktuelle Konfiguration angepasst
4. Client: Auswahl – > Komponent 1-2
5. Server: Komponent 1-2 hinzugefügt, aktuelle Konfiguration angepasst
6. Client: Auswahl: – > nächste Schritt laden
7. Server: Lade Schritt 2
8. Client: Auswahl – > Komponent 2-1
9. Server: Komponent 2-1 hinzugefügt, aktuelle Konfiguration angepasst
10. Client: Auswahl: – > nächste Schritt laden
11. Server: Lade Schritt 3
12. Client: Auswahl: – > Komponent 3-1
13. Server: Komponent 3-1 hinzugefügt, aktuelle Konfiguration angepasst
14. Client: Auswahl – > Komponent 2-1
15. Server: Warnung: Komponente 2-1 und Komponente 3-1 werden aus der Konfiguration entfernt.
16. Client: Auswahl – > Ja
17. Server: Komponente 2-1, Schritt 3 mit Komponente 3-1 aus der aktuellen Konfiguration entfernt.

2.2.2 Szenario 002-2

Globale Einstellungen $\langle configurationCourse = sequence \rangle$

1. Server: Start Konfiguration – > Schritt 1
2. Client: Auswahl – > Komponent 1-1
3. Server: Komponent 1-1 wurde in die aktuelle Konfiguration hinzugefügt
4. Client: Auswahl – > Komponent 1-1
5. Server: Komponent 1-1 wurde aus der aktuellen Konfiguriuon entfernt

2.2.3 Szenario 002-3

Globale Einstellungen $\langle configurationCourse = sequence \rangle$

1. Server: Start Konfiguration – > Schritt 1
2. Client: Auswahl – > Komponent 1-1
3. Server: Komponent 1-1 wurde in die aktuelle Konfiguration hinzugefügt
4. Client: Auswahl – > Komponent 1-2
5. Server: Komponent 1-2 wurde in die aktuelle Konfiguration hinzugefügt
6. Client: Auswahl – > Komponent 1-1
7. Server: Komponent 1-1 wurde aus der aktuellen Konfiguriuon entfernt

2.2.4 Szenario 002-4

Globale Einstellungen $\langle configurationCourse = sequence \rangle$

1. Server: Start Konfiguration – > Schritt 1
2. Client: Auswahl – > Komponent 1-1
3. Server: Komponent 1-1 wurde in die aktuelle Konfiguration hinzugefügt
4. Client: Auswahl – > Komponent 1-2
5. Server: Komponent 1-2 wurde in die aktuelle Konfiguration hinzugefügt
6. Client: Auswahl – > Komponent 1-2
7. Server: Komponent 1-2 wurde aus der aktuellen Konfiguriuon entfernt

2.2.5 Szenario 002-5

Globale Einstellungen $\langle configurationCourse = sequence \rangle$

1. Server: Start Konfiguration – > Schritt 1
2. Client: Auswahl – > Komponent 1-1
3. Server: Komponent 1-1 wurde in die aktuelle Konfiguration hinzugefügt
4. Client: Auswahl: – > nächste Schritt laden
5. Server: Lade Schritt 2
6. Client: Auswahl – > Komponent 2-1
7. Server: Komponent 2-1 wurde in die aktuelle Konfiguration hinzugefügt
8. Client: Auswahl: – > nächste Schritt laden
9. Server: Lade Schritt 3
10. Client: Auswahl – > Komponent 3-1
11. Server: Komponent 3-1 wurde in die aktuelle Konfiguration hinzugefügt
12. Client: Auswahl: – > nächste Schritt laden
13. Server: Error -! Es existiert keiner weiteren Schritt