

Generischer Konfigurator

Gennadi Heimann

29.12.2017

1 Logik Beschreibung

Die Abhängigkeiten im Konfigurator bilden die Regel. Jede Komponente hat eine Ausgangsabhängigkeit (outDependency) und eine Eingangsabhängigkeit (inDependency). Die Ausgangsabhängigkeit bewirkt auf ihre Komponente mit der Regel die in der Abhängigkeit hintergelegt ist. Es gibt zwei Arten (dependencyType) von Abhängigkeiten *< exclude >* und *< require >*. Die *< exclude >* Abhängigkeit schließt die Komponente aus der Konfiguration aus. Die *< require >* Abhängigkeit fordert wiederum, dass die Komponente in die Konfiguration hinzugefügt werden muss. Das Verhalten der Abhängigkeiten wird von Parameter *< visualization >* gesteuert. Der Benutzer des Konfigurators wählt die Komponente innerhalb eines Schrittes und die Logik des Konfigurators prüft die Abhängigkeiten und anhand der Parameter wird weiterer Verlauf gesteuert. In der darauffolgenden Beschreibung werden einzelne Parameter erklärt.

- visualization bei der *< exclude >* Komponenten
 - auto
 - * Die Komponente wird automatisch aus der Konfiguration entfernt. Wenn die ausgeschlossene Komponente weiter auf eine *< exclude >* Komponente verweist, werden diese Komponente recursive ausgeschlossen. Das bedeutet, dass bei der späteren Verlauf werden diese Komponente nicht gezeigt. Bei der *< require >* Komponenten wird dieses Vorgehen nicht angewendet. Die Komponente wird nicht aus der Konfiguration entfernt.
 - marked + selectable
 - * Die IN-Komponente wird visual markiert, dass sie *< exclude >* Komponente ist und der Benutzer hat die Möglichkeit diese Komponente auszuwählen. Wenn der Benutzer wählt diese Komponente aus, wird ein Hinweis dem Benutzer vorgeschlagen, wie er die Abhängigkeiten in seiner Konfiguration lösen kann. Es wird eine andere OUT-Komponente aus dem gleichem Schritt vorgeschlagen. Wenn in diesem Schritt keinen Vorschlag gemacht werden kann wird weiter in dem bergeordneten Schritt weiter gesucht. Bei jedem vorgeschlagenen Komponente wird weiterhin nach Abhängigkeiten geprüft.
 - marked + unselectable
 - * Die Komponente wird visuell markiert aber kann nicht ausgewählt werden. Die Abhängigkeiten werden gleich wie bei der Visualisationsparameter *< auto >* behandelt.
- require
 - auto

- * Die Komponente wird automatisch in die Konfiguration hinzugefügt. Es wird geprüft, ob der IN-Komponente der Abhängigkeit weitere *< require >* oder *< exclude >* Komponente hat. Wenn ja, wird gemäß der Parameter in der Abhängigkeit die Konfiguration angepasst. Wenn die IN-Komponente über mehrere Schritte in der Konfiguration entfernt ist. Der Konfigurator wird, dann in Laufe der Konfiguration bei jedem Schritt dem Benutzer einen Hinweis geben, welche Komponente der Benutzer auswählen kann/muss um zu dem Schritt mit der *< require >* Komponente zu kommen. Die Komponente wird visual markiert, dass sie schon zu der Konfiguration hinzugefügt wurde. Diese Komponente ist auch *< unselectable >*. Bei der Abhängigkeit in gleichem Schritt wird die Komponente ohne jegliche Hinweise zu der Konfiguration hinzugefügt und dementsprechend markiert.
- selectable
 - * Bevor die Komponente zu der Konfiguration hinzugefügt wird, wird dem Benutzer einen Hinweis mit dem Auswahl gegeben. Der Benutzer kann entscheiden ob er die required Komponente zu der Konfiguration hinzufügen will.

Die aktuelle Konfiguration behält die gesamte Information (Abhängigkeiten) von der hinzugefügten Komponente. Diese Information hilft dem Konfigurator die Prüfungen durchführen, wenn der Web-Client Fehler macht. Zum Beispiel, wenn der Web-Client dem Benutzer eine Komponente auszuwählen erlaubt, die von Server als *< unselectable >* oder *< remove >* markiert wurde. In diesem Fall bekommt der Benutzer die Fehlermeldung.

1.1 Auswahlkriterium

Die Anzahl der Komponente die innerhalb eines Schrittes ausgewählt werden können, werden über den Parameter *< SelectionCriterium >* gesteuert. Dieses Kriterium definiert für jeden Schritt die maximale und minimale Anzahl der Komponente, die von dem Benutzer ausgewählt werden können. Das Auswahlkriterium wird nach der Prüfung der Abhängigkeiten durchgeführt. Es werden folgende Fälle unterschieden:

- RequireComponent
 - Wenn die Anzahl der ausgewählten Komponente kleiner als minimale Kriterium. Der Benutzer wird aufgefordert weitere Komponente auszuwählen, bis minimale Kriterium erreicht wird.
- min > countOfComponents**
- RequireNextStep
 - Wenn alle Komponente in dem Schritt ausgewählt sind, wird der Konfigurator zu dem nächsten Schritt führen.

max == countOfComponents

Wenn der minimale Kriterium gleich Null und maximale Kriterium größer Eins ist, kann der Benutzer ohne den Auswahl der Komponente zu dem nächsten wechseln.

min == 0 and max > 1 ist nicht erlaubt

min == 0 and max == 0 ist nicht erlaubt.

- AllowNextComponent

- Wenn dem Benutzer noch weitere Komponente auszuwählen erlaubt. In diesem Fall kann der Benutzer weitere Komponente auszuwählen oder zu dem nächsten Schritt wechseln.

min <= countOfComponents and

max > countOfComponents

- ExcludeComponent

- Wenn der Benutzer mehr Komponente als erlaubt auswählen möchte.

max < countOfComponents

<countOfComponents> = Anzahl der vorher ausgewählten Komponenten.
Die Anzahl der Komponenten wird aus der aktuellen Konfiguration ermittelt

Die Variable *< countOfComponents >* wird vor der Prüfung auf die Auswahlkriterium immer angepasst. Die Ausgewählte Komponente wird zuerst mit aktuellen Konfiguration verglichen.

Wenn die ausgewählte Komponente nicht in der bestehenden Konfiguration existiert, wird *< countOfComponents >* auf 1 inkrementiert. Somit wird temporär die ausgewählte Komponente zu der aktuellen Konfiguration hinzugefügt. Nach dem alle Prüfungen erfolgreich abgelaufen wird diese Komponente zu der aktuellen Konfiguration hinzugefügt.

Wenn die ausgewählte Komponente in der aktuellen Konfiguration gefunden wird. Das bedeutet, dass die Komponente abgewählt wird und aus der Konfiguration entfernt werden muss, deswegen wird *< countOfComponents >* nicht verändert.

Änderung 11.03.2018 bei der RemovedComponent wird selectedComponent - 1 ausgeführt.

1.2 Hinweisfeld zu jeder Aktion

Der Server liefert bei jedem Aktion die Erklärungen zu diesen Aktion, deswegen bei dem Webclient sollte einen Bereich geben wo diese Information präsentiert werden kann.

1.3 Auswahl der Komponente

Der Server bekommt von dem Web-Client die ausgewählte Komponente.

1. Die ausgewählte Komponente wird mit der aktuellen Konfiguration verglichen.
Wenn ausgewählte Komponente in aktueller Konfiguration existiert,
 - JA – > wird die Komponente auf die < *require* > Abhängigkeit geprüft. Wenn andere Komponente die ausgewählte Komponente benötigt,
 - JA – > wird die Regel, die in der Abhängigkeit steht angewendet.
 - NEIN – > wird die Komponente aus der aktuellen Konfiguration entfernt und sind keine weitere Prüfungen notwendig. Bei dieser Prüfung müssen die Komponente und die Schritte auf Konsistenz prüfen. Die Komponente wird als REMOVED_COMPONENT markiert
 - NEIN – > wird die Komponente als ADDED_COMPONENT markiert und an weiteren Prüfungen weitergegeben.
2. Unabhängig von der aktuellen Konfiguration wird die ausgewählte Komponente auf *exclude* > und < *require* > Abhängigkeiten geprüft.
 - die Komponente hat eine < *require* > Eingangsabhängigkeit – >
Die abhängige Komponente werden für weitere Prüfung ermittelt. Der Status wird als REQUIRE_COMPONENT.
 - die Komponente hat keine < *require* > Eingangsabhängigkeit – >
Die ausgewählte Komponente wird ohne weitere Aktionen auszuführen als NOT_REQUIRE_COMPONENT.
 - die Komponente hat eine < *exclude* > Eingangsabhängigkeit – >
Die ausgewählte Komponente wird nach den Regeln in der Abhängigkeit entweder trotz der Ausschluss zu der aktuellen Konfiguration hinzugefügt oder nicht zu der aktuellen Konfiguration hinzugefügt. Der Status wird als EXCLUDE_COMPONENT markiert.
 - die Komponente hat keine < *exclude* > Eingangsabhängigkeit – >
Die ausgewählte Komponente wird ohne weitere Aktionen auszuführen als NOT_EXCLUDE_COMPONENT.
3. Der Status von der Vergleich in der aktuellen Konfiguration und von der Prüfung der Abhängigkeiten wird weiter zu der Prüfung der Auswahlkriterium weitergegeben.
 - ADDED_COMPONENT – >

- REQUIRE_COMPONENT – > Die abhängige Komponente werden je nach Regel in der Abhängigkeit zu der aktuellen Konfiguration hinzugefügt oder dem Regeln entsprechende Behandlung angestossen.
Desweiteren werden alle exclude Komponente innerhalb eines Schrittes und deren Regeln geprüft, Wenn die Regeln schliessen die Komponente aus,
 - * JA – > wird die List von allen innerhalb eines Schrittes < *excluded* > Komponenten, mit der Liste allen nicht ausgewählten Komponenten verglichen. Wenn beide Listen gleich sind,
 - JA – > wird der Status REQUIRE_NEXT_STEP gesetzt. Dadurch wird die Anforderung zu dem nächsten Schritt bei dem Auswahl der Komponente, der alle andere Komponente innerhab des Schrittes ausschliesst, erkannt.
 - NEIN – > wird eine standartmssige Überprüfung der Auswahlkriterium gemacht. (siehe Auswahlkriterium)
 - * NEIN – > Definition folgt
- NOT_REQUIRE_COMPONENT – > Es werden alle exclude Komponente innerhalb eines Schrittes und deren Regeln geprüft, Wenn die Regeln schliessen die Komponente aus,
 - * JA – > wird die List von allen innerhalb eines Schrittes < *excluded* > Komponenten, mit der Liste allen nicht ausgewählten Komponenten verglichen. Wenn beide Listen gleich sind,
 - JA – > wird der Status REQUIRE_NEXT_STEP gesetzt. Dadurch wird die Anforderung zu dem nächsten Schritt bei dem Auswahl der Komponente, der alle andere Komponente innerhab des Schrittes ausschliesst, erkannt.
 - NEIN – > wird eine standartmssige Überprüfung der Auswahlkriterium gemacht. (siehe Auswahlkriterium)
 - * NEIN – > Definition folgt
- EXCLUDE_COMPONENT – > setzen der Status bei der Vergleich der aktuellen Konfiguration zu der NOT_ALLOWED_COMPONENT, wenn der Regel in der Abhängigkeit das erlaubt. Die Komponente wird nicht zu der aktuellen Konfiguration hinzugeügt. Der Staus der Auswahlkriterium wird auf EXCLUDED_COMPONENT gesetzt.
- NOT_EXCLUDE_COMPONENT – >
- REMOVED_COMPONENT – > SelectionCriterium muss gepueft werden.

1.4 Status

- SelectedComponent beschreibt der Status der ausgewhlte Komponente

- REMOVED_COMPONENT beschreibt die ausgewählte Komponente, die zweites Mal ausgewählt wurde. Bei erstem Mal wird die Komponente zu der Konfiguration hinzugefügt, beim zweiten Mal wird die Komponente aus der Konfiguration entfernt. (Abzuwählen)
 - ADDED_COMPONENT beschreibt die Komponente die erstes Mal ausgewählt wurde. Diese Komponente wird zu der Konfiguration hinzugefügt.
 - ERROR_COMPONENT beschreibt die Komponente bei dem nach der Auswahl einen allgemeinen Fehler passiert.
 - NOT_ALLOWED_COMPONENT beschreibt die Komponente die aufgrund der Abhängigkeit nicht zu der Konfiguration hinzugefügt werden kann.
- SelectionCriterion beschreibt den Status der Komponente vor der Auswahl.
 - siehe Auswahlkriterium