Administrator für Generischer Konfigurator

Gennadi Heimann

29.12.2017

1 Beschreibung

Der Administrator für den generischen Konfigurator übernimmt die gesamte Einstellung. Die Logik des Konfigurators besteht aus drei Komponenten.

- Schritt
- Komponente
- Abhängigkeit

2 Logik Beschreibung

Die Abhngigkeiten im Konfigurator bilden die Regel. Jede Komponente hat eine Ausgangsabhngigket (outDependency) und eine Eingangsabhngigket (inDependency). Die Ausgangsabhngigket bewirkt auf ihre Komponente mit der Regel die in der Abhngigkeit hintergelegt ist. Es gibt zwei Arten (dependencyType) von Abhngigkeiten $< exclude >$ und $< require >$. Die $< exclude >$ Abhngigkeit schließt die Komponente aus der Konfiguration aus. Die $< require >$ Abhngigkeit fordert wiederum, dass die Komponente in die KOnfiguration hinzugefügt werden muss. Das Verhalten der Abhïgigkeiten wird von Parameter $< visualization >$ gesteuert. Der Benutzer des Konfigurators whlt die Komponente innerhalb eines Schrittes und die Logik das Konfigurators prüft die Abhngigkeiten und anhand der Parameter wird weitere Verlauf gesteuert. In der darauffolgende Beschreibung werden einzelne Parameter erklrt.

- visualization bei der $< exclude >$ Komponenten
  - auto
    - Die Komponente wird automatisch aus der Konfiguration entfernt.Wenn die ausgeschlossene Komponente weiter auf eine $< exclude >$ Komponente verweißt, werden diese Komponente recursive ausgeschlossen. Das bedeutet, dass bei der spterem 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 merkiert, dass sie $< exclude >$ Komponente ist und der Benutzer hat die Möglichkeit diese Komponente auszuwhlen. Wenn der Benutzer whlt diese Komponente aus, wird ein Hinweis dem Benutzer vorgeschlagen, wie er die Abhngigkeiten 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 übergeordneten Schritt weiter gesucht. Bei jedem vorgeschlagenen Komponente wird weiterhin nach Abhngigkeiten geprft.

  - marked + unselectable
    * Die Komponente wird visuel markiert aber kann nicht ausgewhlt werden. Die Abhngigkeiten werden gleich wie bei der Visualizationparameter $< auto >$ behandelt.

- require
  - auto
    * Die Komponente wird automatisch in die Konfiguration hinzugefügt. Es wird geprüft, ob der IN-Komponente der Abhngigkeit weitere $< require >$ oder $< exclude >$ Komponente hat. Wenn ja, wird gemß der Parameter in der Abhngigkeit die Konfiguration angepasst. Wenn die IN-Koponente über mehrere Schritt 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 auswhlen kann/muss um zu dem Schritt mit der $< require >$ Komponente zu kommen. Die Komponente wird visual markiert, dass sie schon zu der Konfiguration hinzugefugt wurde. Diese Komponente ist auch $< unselectable >$. Bei der Abhngigkeit 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 behlt die gesamte Information (Abhngigkeiten) 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 auszuwhlen erlaubt, die von Server als $< unselecteble >$ oder $< remove >$ markiert wurde. In diesem Fall bekommt der Benutzer die Fehlermeldung.

2.1 Auswahlkriterium

Die Anzahl der Komponente die innerhalb eines Schrittes ausgewhlt werden können, werden über den Parameter $< SelectionCriterum >$ gesteuert. Dieses Kriterium definiert für jeden Schritt die maximale und minimale Anzahl der Komponente, die von dem Benutzer ausgehlt werden können. Das Auswahlkriterium wird nach demPrüfung der Abhngigkeiten durchgeführt. Es werden follgende Flle unterschieden:

- RequireComponent

  - Wenn die Anzahl der ausgewhlten Komponente kleiner als minimale Kriterium. Der Benutzer wird aufgefordert weiter Komponente auszuwhlen, bis minimale Kriterium erreicht wird.
    min > countOfComponents

- RequireNextStep
  - Wenn alle Komponente in dem Schritt ausgewhlt sind, wird der Konfigurator zu dem nchstem 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 nchsten wechseln.
    min == 0 and max > 1 ist nicht erlaubt
    min == 0 and max == 0 ist nicht erlaubt.

- AllowNextComponent
  - Wenn dem Benutzer noch weitere Komponente auszuwhlen erlaubt. In diesem Fall kann der Benutzer weitere Komponente auszuwahlen oder zu dem nchstem Schritt wechseln.
    min <= countOfComponents and
    max > countOfComponents

- ExcludeComponent
  - Wenn der Benutzer mehr Komponente als erlaubt auswhlen möchte.
    max < countOfComponents

<**countOfComponents**> = Anzahl der vorher ausgewhlten Komponenten. Die Anzahl der Komponenten wird aus der aktuellen Konfiguraion ermittelt

Die Variable $< countOfComponents >$ wird vor der Prüfung auf die Auswahlkriterium immmer angepasst. Die Ausgewhlte Komponnete wird zuerst mit aktuellen Konfiguration verglichen.

Wenn die ausgewhlte Komponente nicht in der bestehenden Konfiguration exestiert, wird $< countOfComponents >$ auf 1 inkrementiert. Somit wird temporr die ausgewhlte Komponnte zu der aktuellen Konfiguration hinzugefügt. Nach dem alle Prüfungen erfolgreich abgelaufen wird diese Komponnete zu der aktuellen Konfiguration hinzugefügt.

Wenn die ausgewhlte Komponente in der aktuellen Konfiguration gefunden wird. Das bedeutet, dass die Komponnete abgewhlt wird und uas der Konfiguration entfernt werden muss, deswegen wird $< countOfComponents >$ nicht verndert.

Aenderung 11.03.2018 bei der RemovedComponent wird selectedComponent - 1 ausgeführt.

2.2 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.

2.3 Auswahl der Komponente

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

1. Die ausgewählte Komponente wird mit der aktuelle Konfiguration vergliechen.
   Wenn ausgewählte Komponente in aktueller Konfiguration exestiert,
   - 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 prufen. 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 weieter Prüfung ermittelt. Der Status wird als REQUIRE_COMPONENT.
   - die Komponente hat keine $<$ *require* $>$ Eingangsabhängigkeit $->$ Die ausgewählte Komponente wird ohne weiter Aktionen auszuführen als NOT_REQUIRE_COMPONENT.
   - die Komponente hat eine $<$ *exclude* $>$ Eingangsabhängigkeit $->$ Die ausgewählte Komponente wird nach der 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 weiter Aktionen auszuführen als NOT_EXCLUDE_COMPONENT.

3. Die 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 nächstem Schritt bei dem Auswahl der Komponente, der alle andere Komponente innerhab des Schrittes ausschliesst, erkannt.
         · NEIN − > wird eine standartmässige 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 nächstem Schritt bei dem Auswahl der Komponente, der alle andere Komponente innerhab des Schrittes ausschliesst, erkannt.
         · NEIN − > wird eine standartmässige 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 geprueft werden.

2.4 Status

- SelectedComponent beschreibt der Status der ausgewhlte Komponente
  - REMOVED_COMPONENT beschreibt der ausgewhlte Komponente, die zweites Mal ausgewhlt wurde. Bei ersten Mal wird die Komponente zu der Konfiguration hinzugefügt, beim zweiten Mal wird die Komponente aus der Konfiguration entfernt. (Abzuwhlen)
  - ADDED_COMPONENT beschreibt die Komponente die erstes Mal ausgewhlt wurde. Diese Komponente wird zu der Konfiguration hinzugefügt.
  - ERROR_COMPONENT beschreibt die Komponente bei dem nach der Auswhl einen allgemeinen Fehler passiert.
  - NOT_ALLOWED_COMPONENT beschreibt die Komponente die aufgrund der Abhängigkeit nicht zu der Konfiguration hinzugefügt werden kann.
- SelectionCriterium beschreibt der Status der Komponente vor dem Auswahl.
  - siehe Auswahlkriterium