ELO Flows Development

Aufgaben

Inhaltsverzeichnis

Implementierungsaufgaben

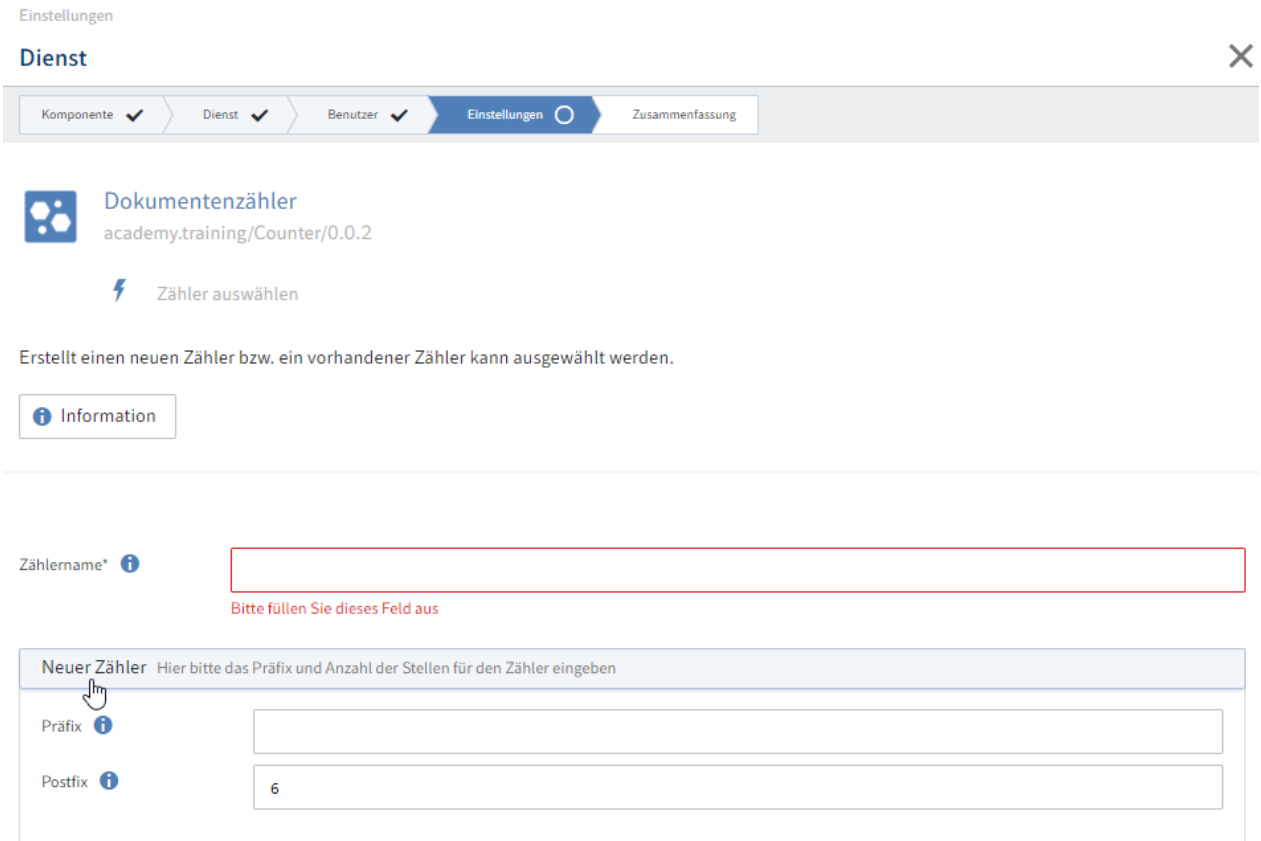
Dokumentenzähler

Wir möchten jetzt einen Zähler (fachliche Identifikation eines Dokumentes) für die im Repository abgelegten Dokumente implementieren. Der Zähler garantiert, dass die Dokumente eines bestimmten Typs (z. B. festgelegt durch den BS SOL_TYPE oder die Metadatenmaske) eine eindeutige fortlaufende Kennung bekommen. Als Beispiel kann man eine Vertrags- oder Rechnungsnummer nennen.

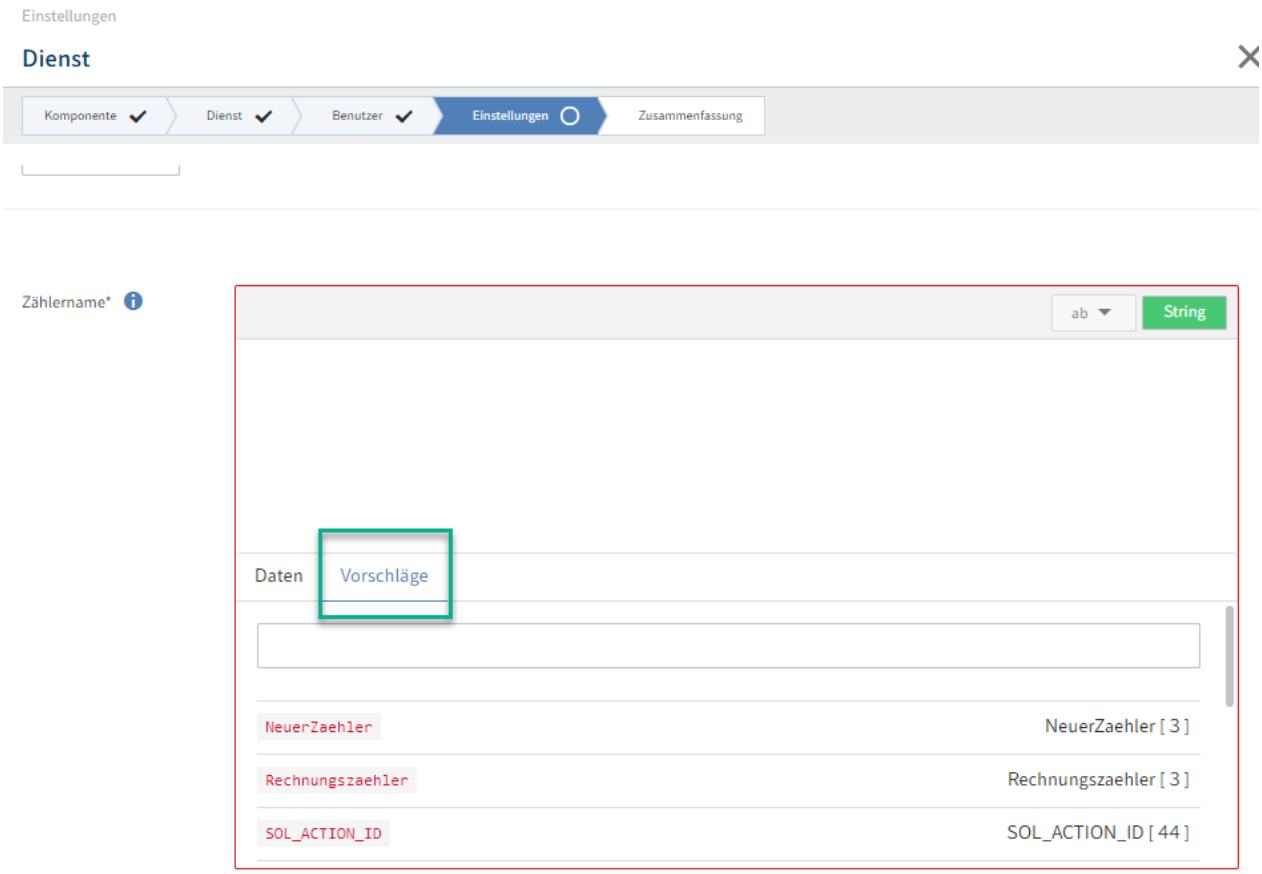
Anforderungen (allgemein)

- Auswahl eines Zählers (ein Zähler kann bereits vorhanden sein)
- Definition/Eingabe eines neuen Zählers (Prefix, Postfix - Anzahl der Stellen)
- Erstellung eines neuen Zählers
- Rückgabe des aktuellen Zählerstandes in weiteren Komponenten
- Bereitstellung eines Triggers (REST)

Eine mögliche Anordnung der grafischen Komponenten im Tab *Einstellungen*.



Auswahlliste in den Vorschlägen.

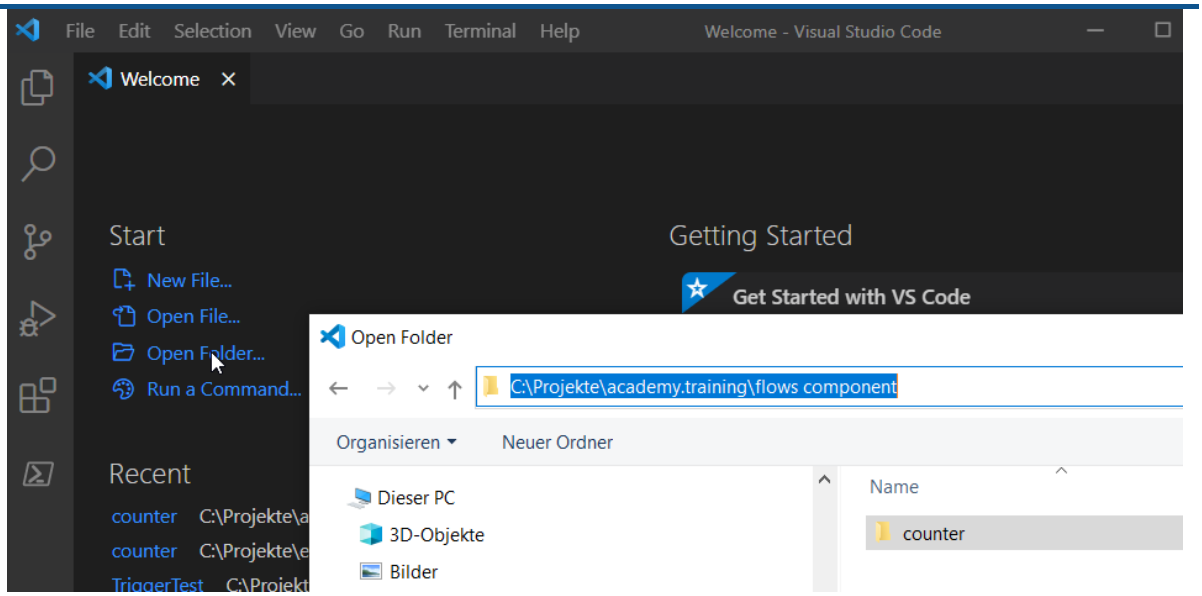


Umsetzung der Anforderungen

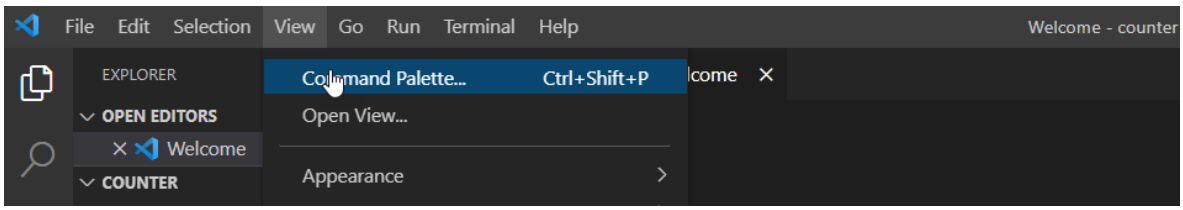
An dieser Stelle wird zum einen der Entwicklungsprozess zum anderen die Implementierung der Lösung skizziert. Die Anforderungen werden wir mithilfe des uns zur Verfügung stehenden Flows-Framework im Visual Studio Code durchführen.

Framework starten

1. Als ersten Schritt starten wir VS Code.
2. Öffnen Sie den für die Umsetzung vorbereiteten Projektordner (im Dateisystem).



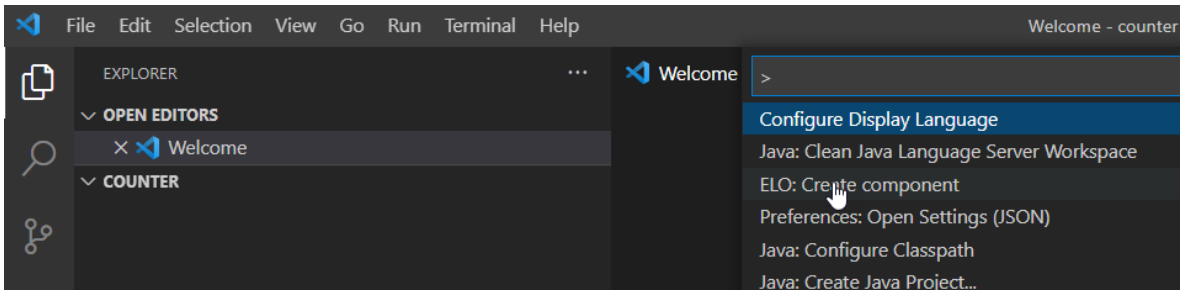
3. Weiterhin erstellen wir die initiale Projektstruktur wie in den nachfolgenden Abbildungen zu sehen ist.



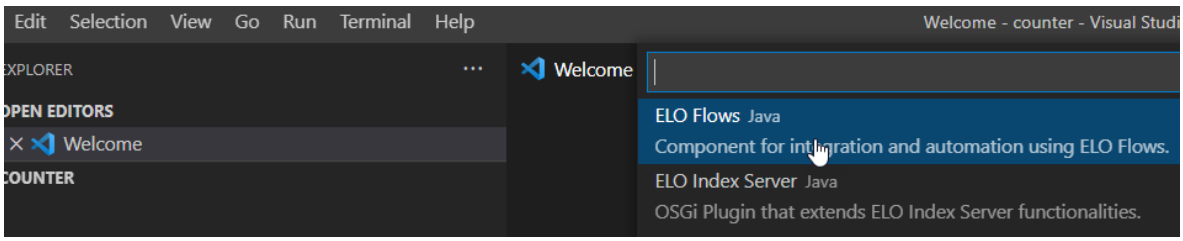
4. Wählen Sie die VS Code Befehlspalette aus. Benutzen Sie hierfür den Tastaturbefehl STRG + SHIFT + P.

   Alternative1: Drücken Sie F1.

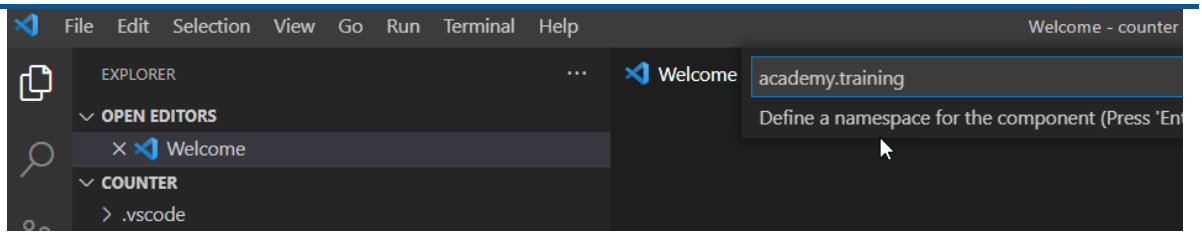
   Alternative2: *View > Command Palette...*.



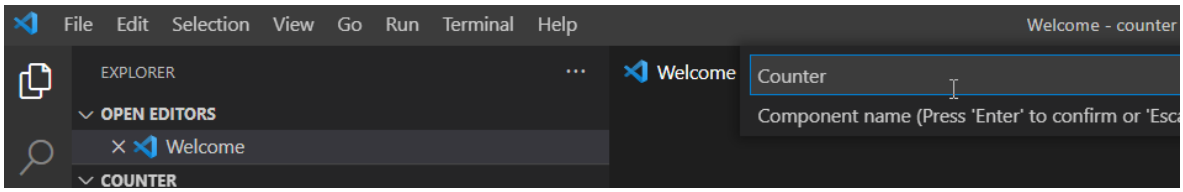
5. Wählen Sie *ELO: Create Component* aus.



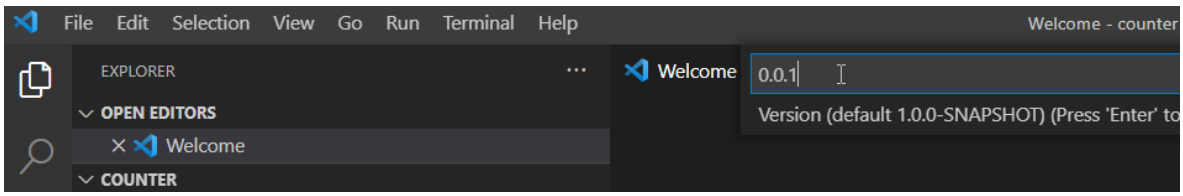
6. Wählen Sie *ELO Flows:Java* aus.



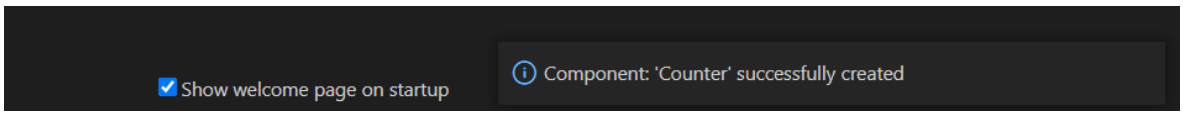
7. Geben Sie den Paketnamen ein und bestätigen Sie die Eingabe mit ENTER.



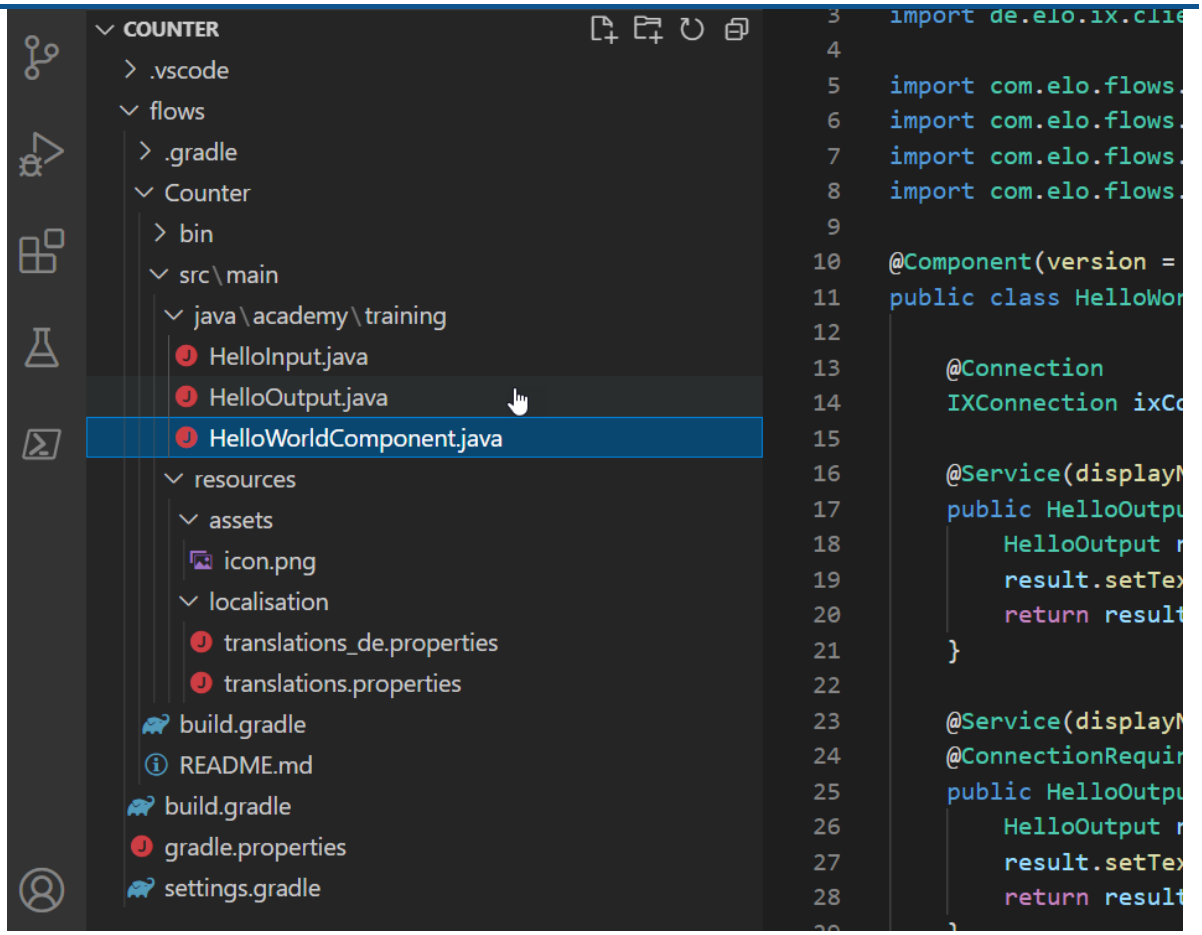
8. Geben Sie den Komponentennamen ein und bestätigen Sie die Eingabe mit ENTER.



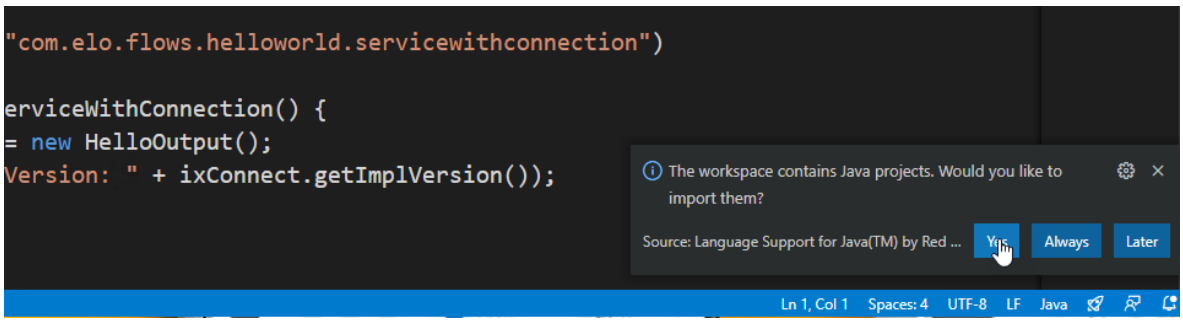
9. Geben Sie die Versionsnummer ein und bestätigen Sie die Eingabe mit ENTER.



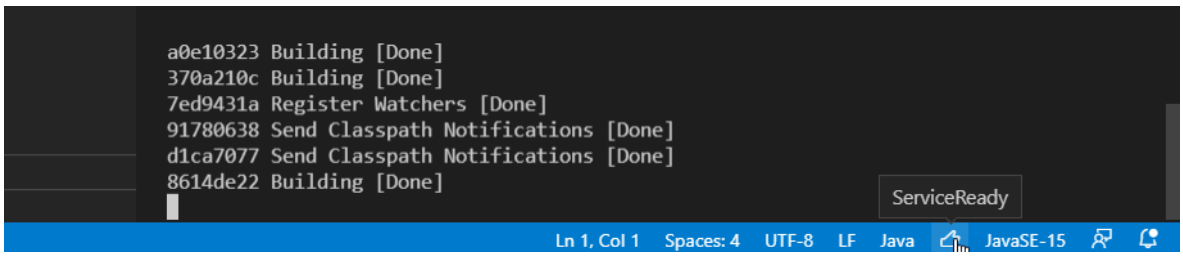
Wenn das Framework gestartet und das initiale Projekt angelegt wurde, dann erhalten Sie eine Bestätigungsmeldung (rechts unten im VSC).



10. Wählen Sie jetzt in der angelegten Projektstruktur eine beliebige Java-Klasse aus.



11. Bestätigen Sie den Vorgang wie unten abgebildet.



12. Warten Sie kurz, bis die Initialisierung des Java-Projektes erfolgreich abgeschlossen ist.

> **Information**
>
> Bei der ersten Initialisierung kann dies durchaus einige Sekunden in Anspruch nehmen.

Projektstruktur anpassen

ELO hat in einer neu angelegten Flows-Komponente eine Beispielkomponente implementiert. Sie können diese für neue Anforderungen anpassen oder durch vorheriges Löschen komplett neu aufsetzen. Im folgenden Beispiel gehen wir einen Mittelweg.

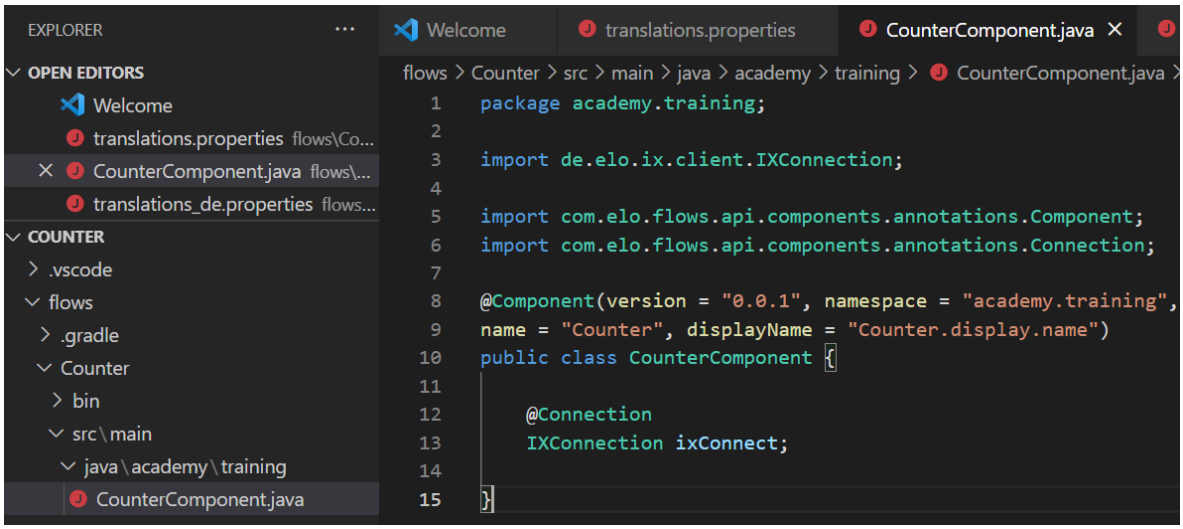
1. Löschen Sie den Inhalt der Dateien im Ordner *localisation*:

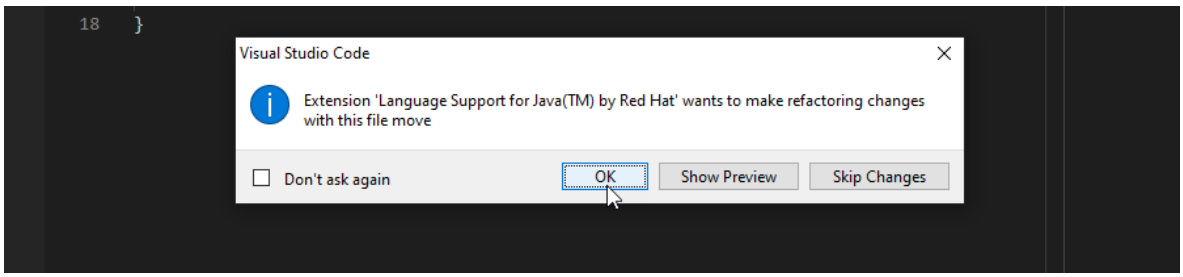
   > **Achtung**
   >
   > Löschen Sie keine Dateien. Löschen Sie ausschließlich die Inhalte der Dateien.

   translations_de.properties und *translations.properties*.

2. Löschen Sie die Java-Klassen im Ordner *java\academy\training*:

   `HelloInput.java` und `HelloOutput.java`





3. Passen Sie die Java-Klasse `HelloWorldComponent.java` an:

   Nutzen Sie hierzu die vom VS Code angebotene Refactoring-Möglichkeiten

translations_de.properties

flows > Counter > src > main > resources > localisation > translations_de.properties

```
1  Counter.display.name=Dokumentenzähler
```

translations.properties

flows > Counter > src > main > resources > localisation > translations.properties

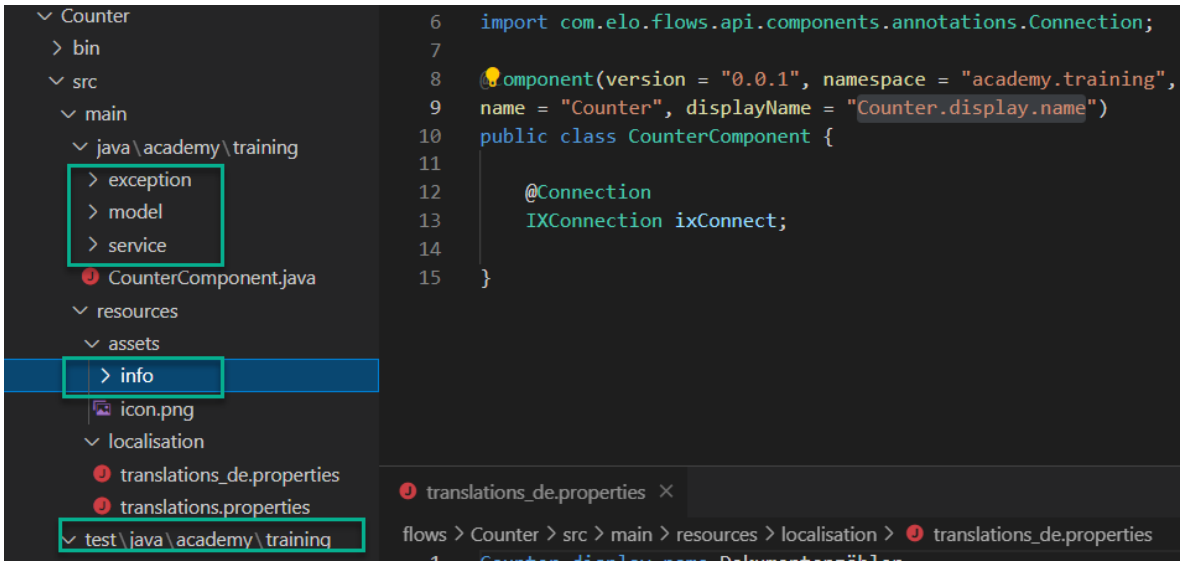
```
1  Counter.display.name=Document Counter
```

4. Tragen Sie folgendes in den Lokalisierungsdateien im Ordner *localisation* ein:

   `translations\_de.properties` und `translations.properties`

   Schlüssel: `Counter.display.name`

   Werte: `Dokumentenzähler` und `Document Counter`.



5. Erweitern Sie die Projektstruktur um weitere Ebenen.

Eine Beispielumsetzung dieser Anforderung können Sie im Anhang finden.

ELOas Modul

In dieser Aufgabe werden wir die Funktionalität aus dem ELOas Modul (teilweise) nachimplementieren.

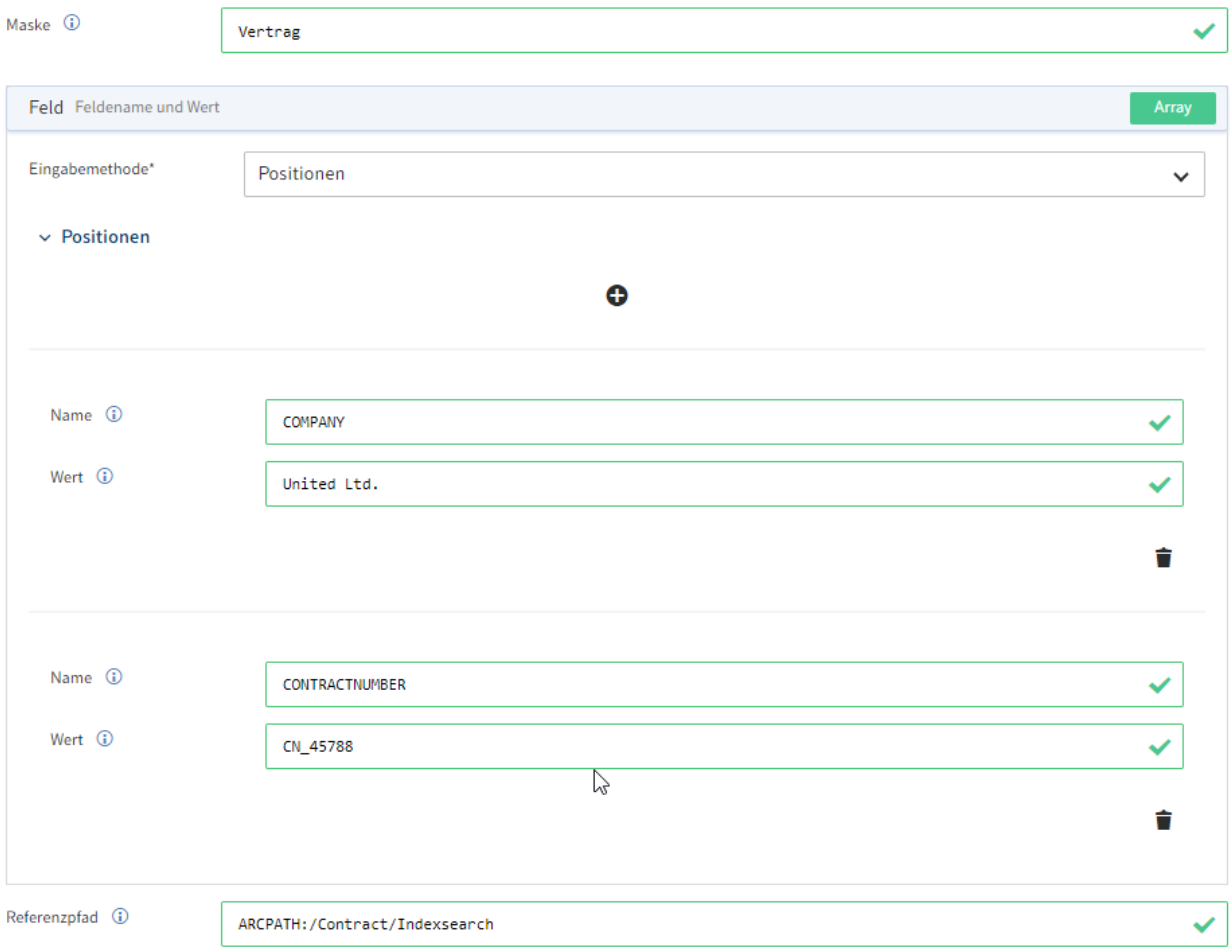
Wir wollen eine Suche (Indexsuche) auslösen, die als Suchparameter die Metadatenmaske und Gruppenfelder definiert.

Die gefundenen Objekte werden in eine angegebene Repository-Struktur referenziert.

Anforderungen (allgemein)

- Angabe einer Metadatenmaske
- Auswahl einer Metadatenmaske aus der Vorschlagsliste
- Angabe von mehreren Feldern, d.h. Name und der Suchwert
- Angabe des Ablagepfades für die Referenzen der gefundenen Objekte
- Bereitstellung eines Triggers (REST)

Die Konfigurationsoberfläche im Tab *Einstellungen*.



Teilnehmeraufgabe (optional)

Diese Aufgabe wird nur bearbeitet, wenn von den Teilnehmern Vorschläge zur Umsetzung ganz bestimmter Anforderung gemacht werden.