ELO Flows Development

Tasks

Table of contents

Implementation tasks

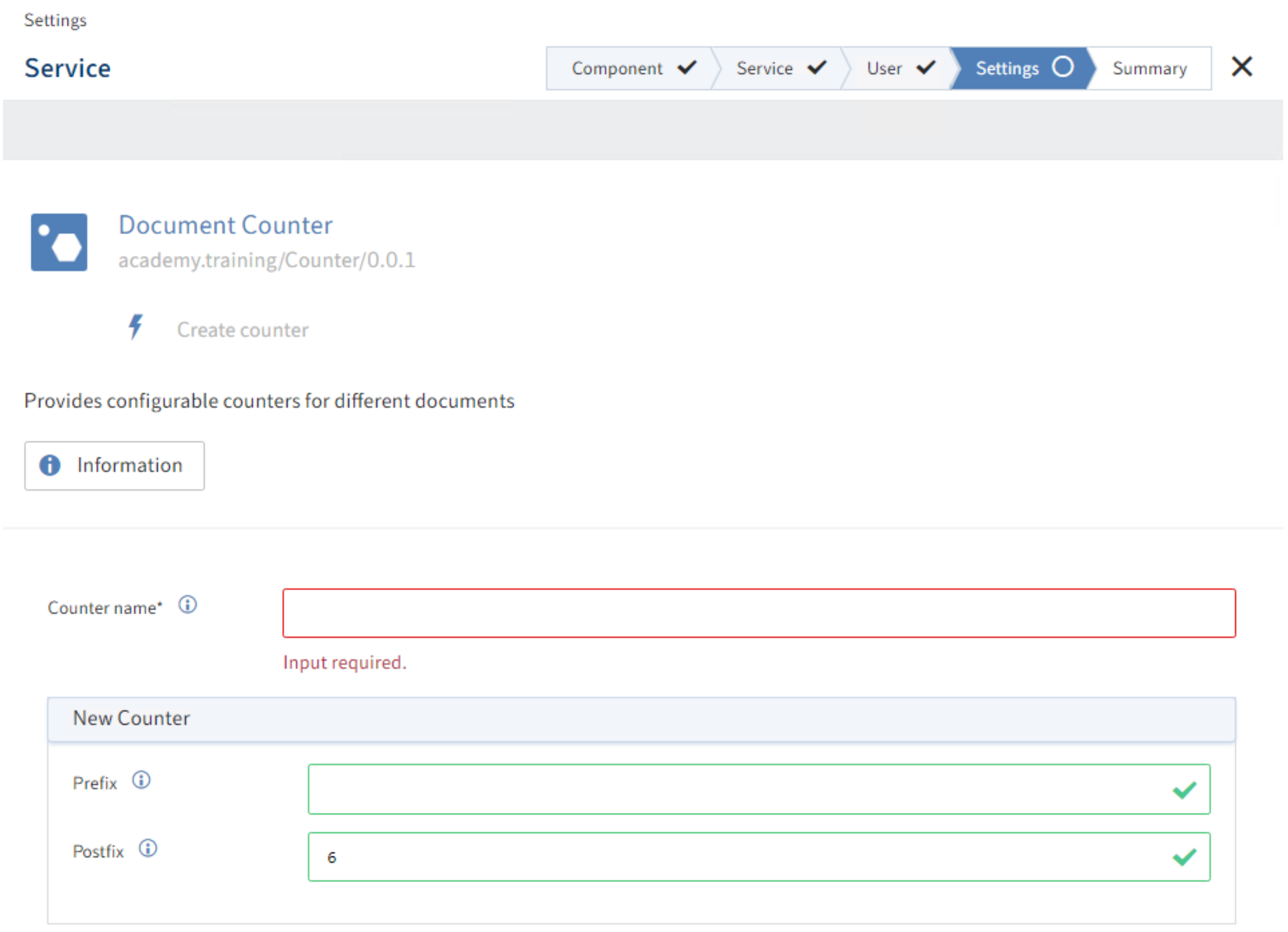
Document counter

We now want to implement a counter (technical identification for a document) for the documents filed to the repository. The counter guarantees that the documents of a specific type (e.g. defined by the ELO Business Solution SOL_TYPE or the metadata form) are assigned a unique serial identifier. One example is a contract or invoice number.

Requirements (general)

- Select a counter (a counter can already exist)
- Define/input a new counter (prefix, postfix, number of digits)
- Create a new counter
- Return the current count in other components
- Provide a trigger (REST)

A possible arrangement of the graphical components on the *Settings* tab.



Document Counter
academy.training/Counter/0.0.1

Create counter

Provides configurable counters for different documents

Counter name*

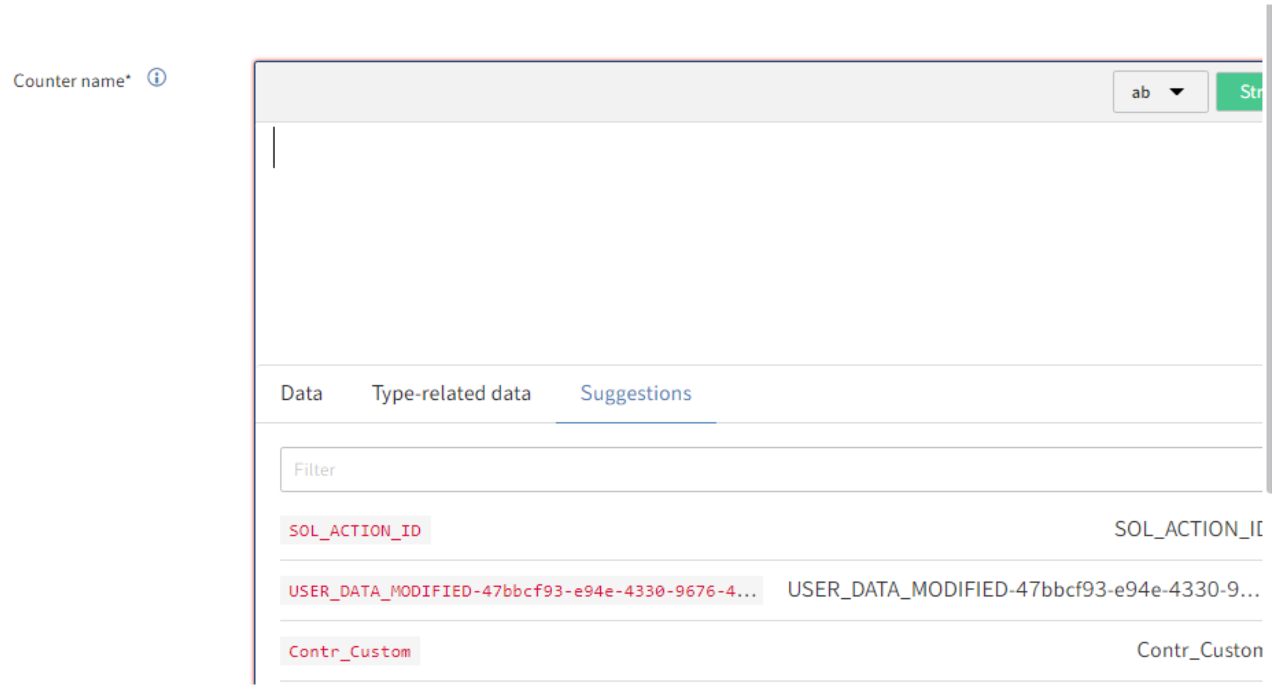
Input required.

New Counter

Prefix

Postfix: 6

Selection list in the suggestions.



SOL_ACTION_ID SOL_ACTION_II

USER_DATA_MODIFIED-47bbcf93-e94e-4330-9676-4... USER_DATA_MODIFIED-47bbcf93-e94e-4330-9...

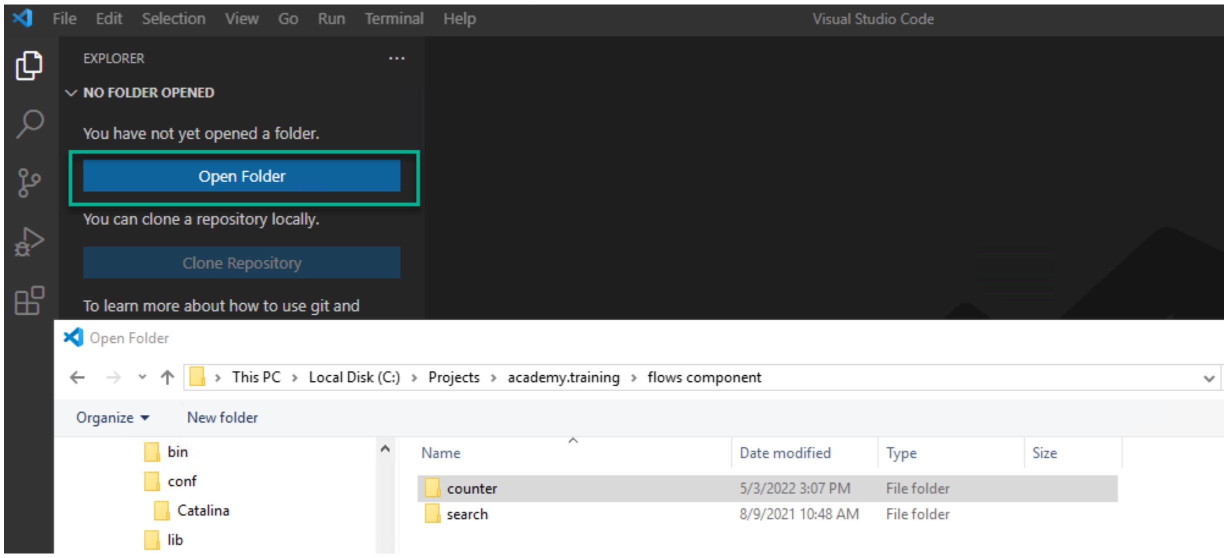
Contr_Custom Contr_Custon

Implementing requirements

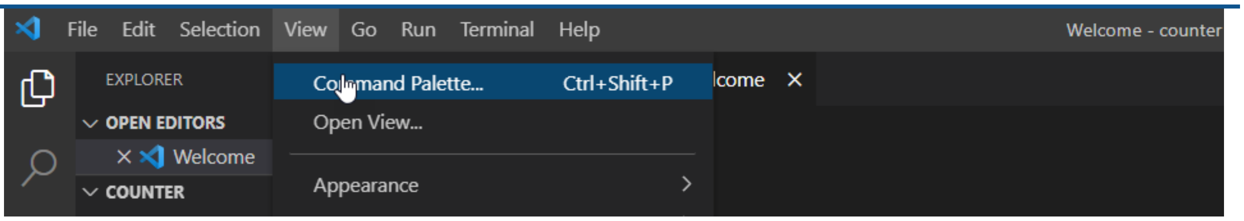
We will outline the development processes as well as solution implementation here. We will go through the steps using the Flows framework available to us in Visual Studio Code.

Start framework

1. First, we start VS Code.
2. Open the prepared project folder (in the file system).


File Edit Selection View Go Run Terminal Help
Visual Studio Code
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
You can clone a repository locally.
Clone Repository
To learn more about how to use git and
Open Folder
This PC › Local Disk (C:) › Projects › academy.training › flows component
Organize
New folder
bin
conf
Catalina
lib
Name
Date modified
Type
Size
counter
5/3/2022 3:07 PM
File folder
search
8/9/2021 10:48 AM
File folder

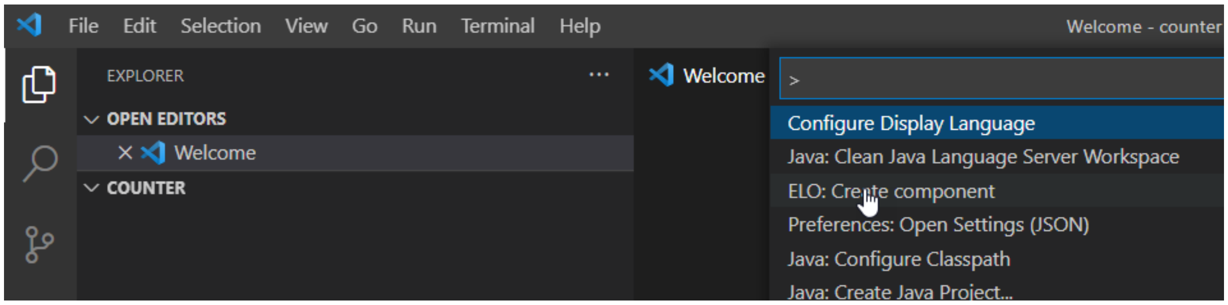
3. We will also create the initial project structure as seen in the following images.



4. Select the VS Code Command Palette. Use the keyboard shortcut CTRL + SHIFT + P.

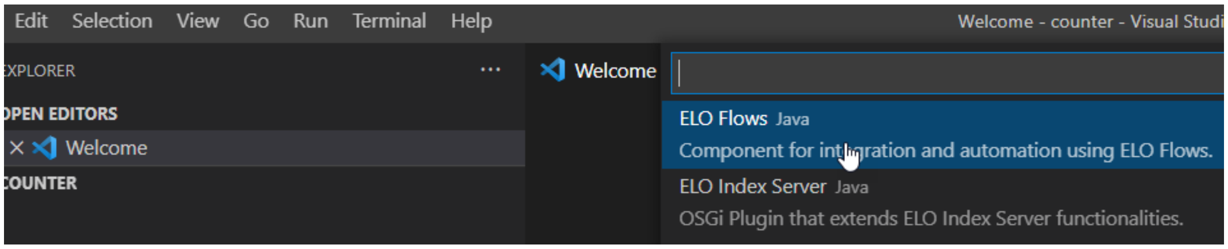
   Alternative1: Press F1.

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



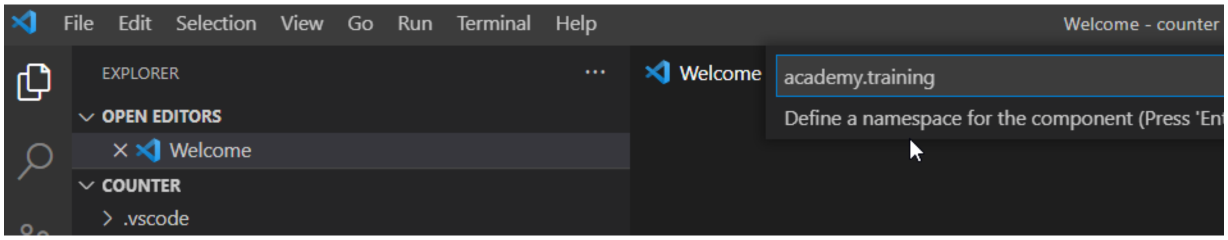
Configure Display Language
Java: Clean Java Language Server Workspace
ELO: Create component
Preferences: Open Settings (JSON)
Java: Configure Classpath
Java: Create Java Project...

5. Select *ELO: Create Component*.



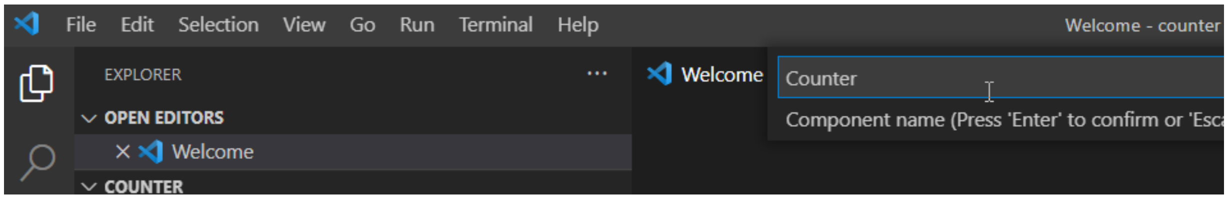
ELO Flows Java
Component for integration and automation using ELO Flows.
ELO Index Server Java
OSGi Plugin that extends ELO Index Server functionalities.

6. Select *ELO Flows Java*.



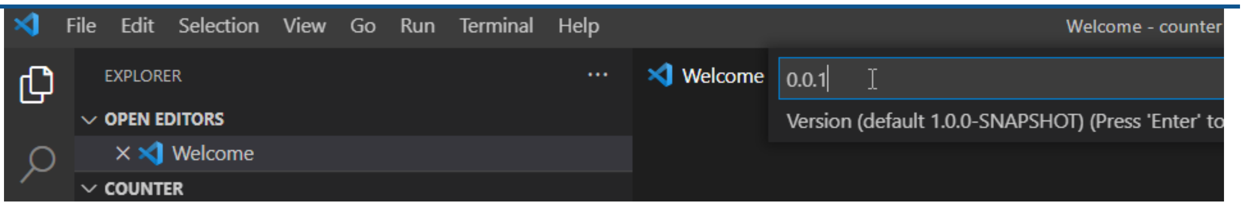
academy.training
Define a namespace for the component (Press 'En

7. Enter the package name and confirm with ENTER.



Counter
Component name (Press 'Enter' to confirm or 'Esca

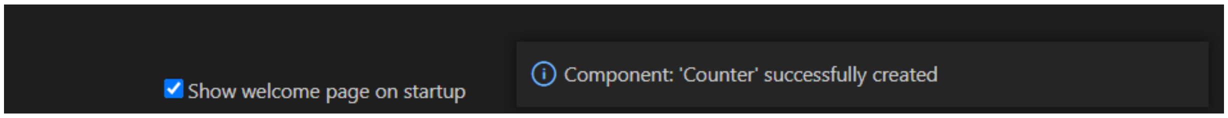
8. Enter the component name and confirm with ENTER.



0.0.1

Version (default 1.0.0-SNAPSHOT) (Press 'Enter' to

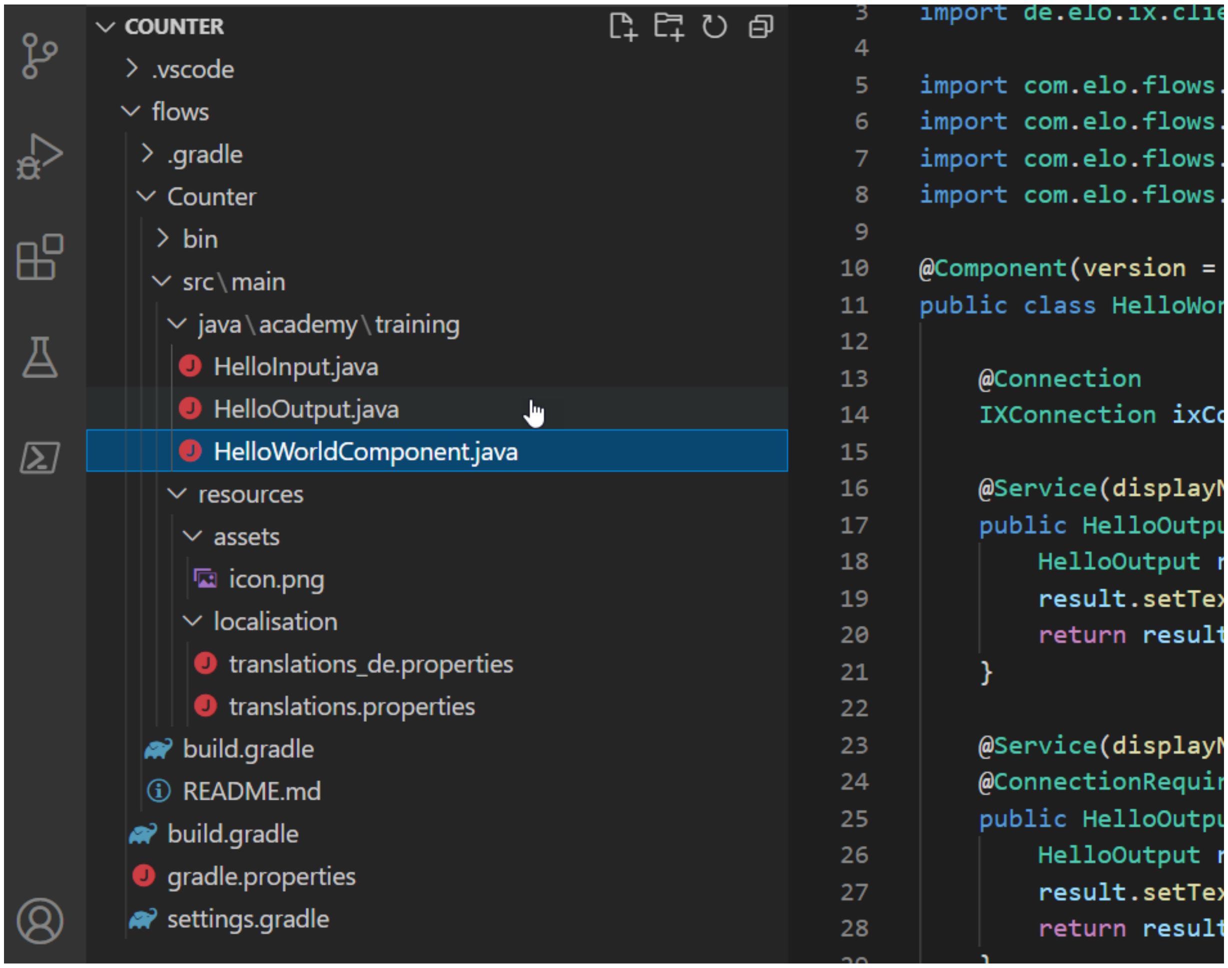
9. Enter the version number and confirm with ENTER.



[✓] Show welcome page on startup

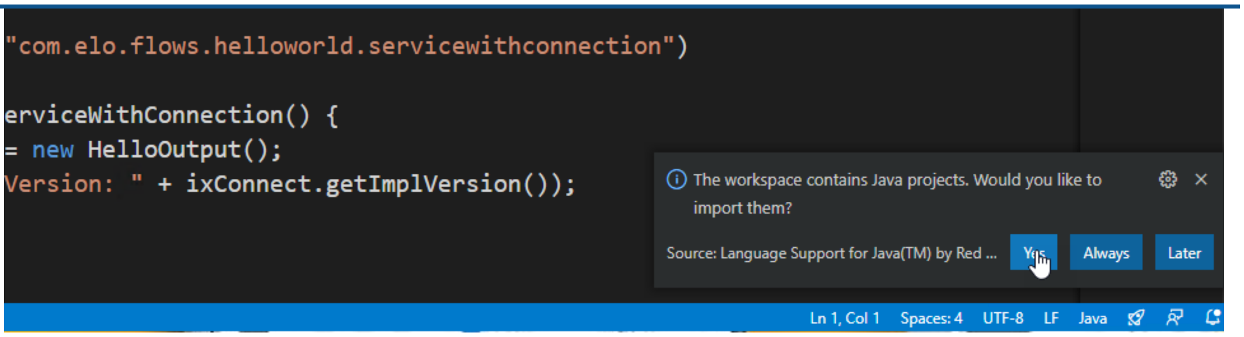
Component: 'Counter' successfully created

Once the framework has been started and the initial project is created, you will see a confirmation message the bottom right of VSC.



```
import de.elo.ix.clie

import com.elo.flows.
import com.elo.flows.
import com.elo.flows.
import com.elo.flows.

@Component(version =
public class HelloWor

    @Connection
    IXConnection ixCo

    @Service(displayN
    public HelloOutpu
        HelloOutput r
        result.setTex
        return result
    }

    @Service(displayN
    @ConnectionRequir
    public HelloOutpu
        HelloOutput r
        result.setTex
        return result
```

10. In the created project structure, select any Java class.

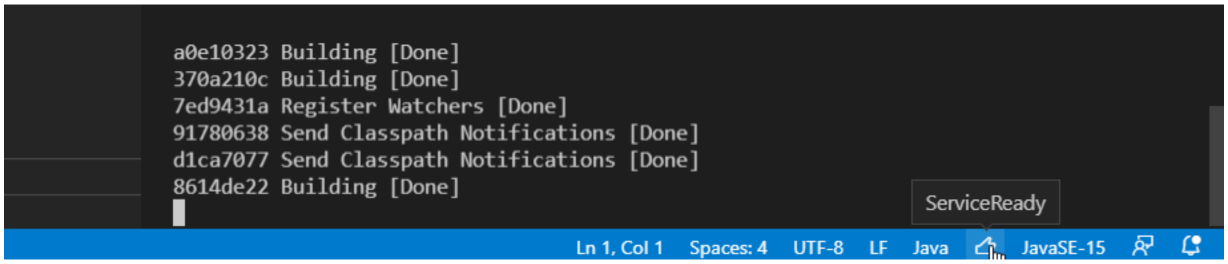


```
"com.elo.flows.helloworld.servicewithconnection")

erviceWithConnection() {
= new HelloOutput();
Version: " + ixConnect.getImplVersion());
```

The workspace contains Java projects. Would you like to import them?

Source: Language Support for Java(TM) by Red ...

11. Confirm as shown below.



```
a0e10323 Building [Done]
370a210c Building [Done]
7ed9431a Register Watchers [Done]
91780638 Send Classpath Notifications [Done]
d1ca7077 Send Classpath Notifications [Done]
8614de22 Building [Done]
```

12. Wait a moment until the Java project has been initialized successfully.

Information

The first time you initialize a project may take a few seconds.

Customizing the project structure

ELO has implemented an example component in a newly created ELO Flows component. You can tailor it to new requirements or delete it and create an entirely new one. In the following example, we take a middle course.

1. Delete the contents of the files in the *localization* folder:

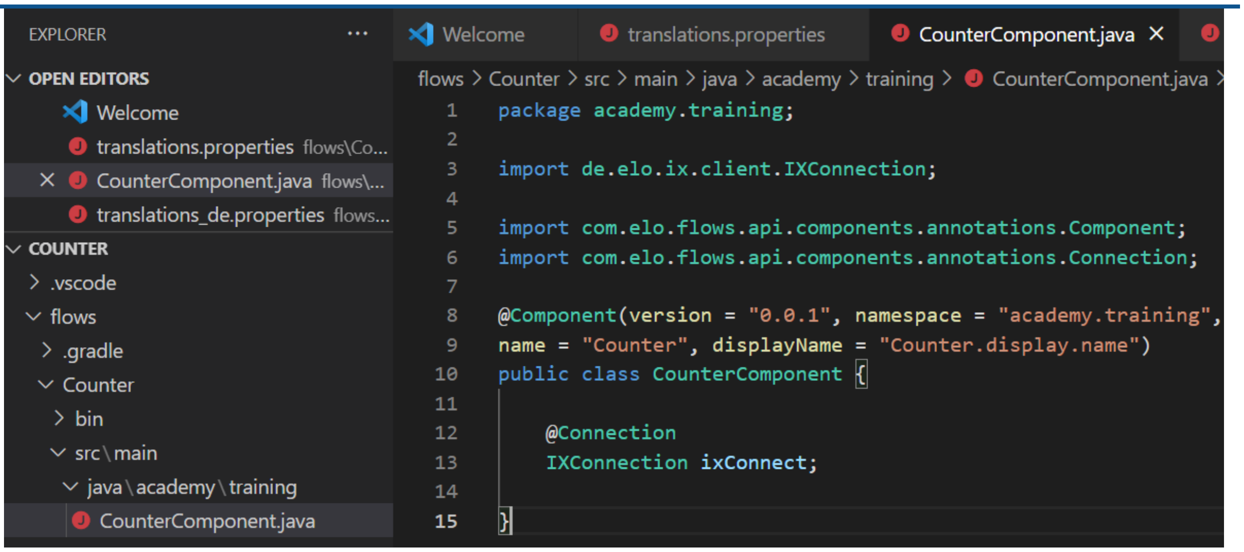
   Important

   Do not delete any files. Only delete the contents of the files.

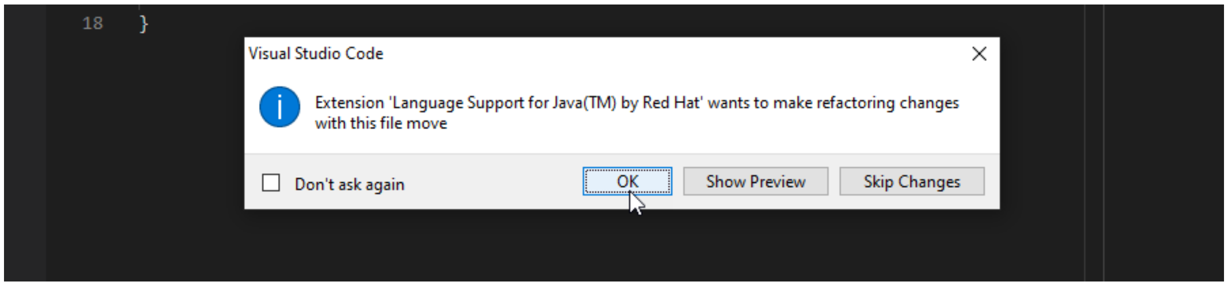
   translations_de.properties and *translations.properties*.

2. Delete the Java classes in the folder *java\academy\training*:

   `HelloInput.java` and `HelloOutput.java`


EXPLORER
OPEN EDITORS
Welcome
translations.properties flows\Co...
CounterComponent.java flows\...
translations_de.properties flows...
COUNTER
.vscode
flows
.gradle
Counter
bin
src\main
java\academy\training
CounterComponent.java
flows > Counter > src > main > java > academy > training > CounterComponent.java

```java
package academy.training;

import de.elo.ix.client.IXConnection;

import com.elo.flows.api.components.annotations.Component;
import com.elo.flows.api.components.annotations.Connection;

@Component(version = "0.0.1", namespace = "academy.training",
name = "Counter", displayName = "Counter.display.name")
public class CounterComponent {

    @Connection
    IXConnection ixConnect;

}
```

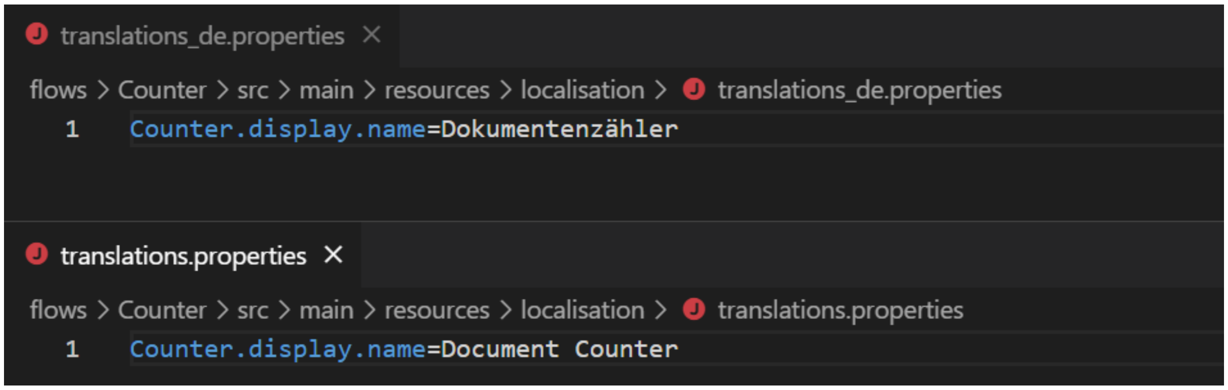


Visual Studio Code

Extension 'Language Support for Java(TM) by Red Hat' wants to make refactoring changes with this file move

3. Modify the Java class `HelloWorldComponent.java`:

   To do so, use the refactoring options provided by VS Code


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

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

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

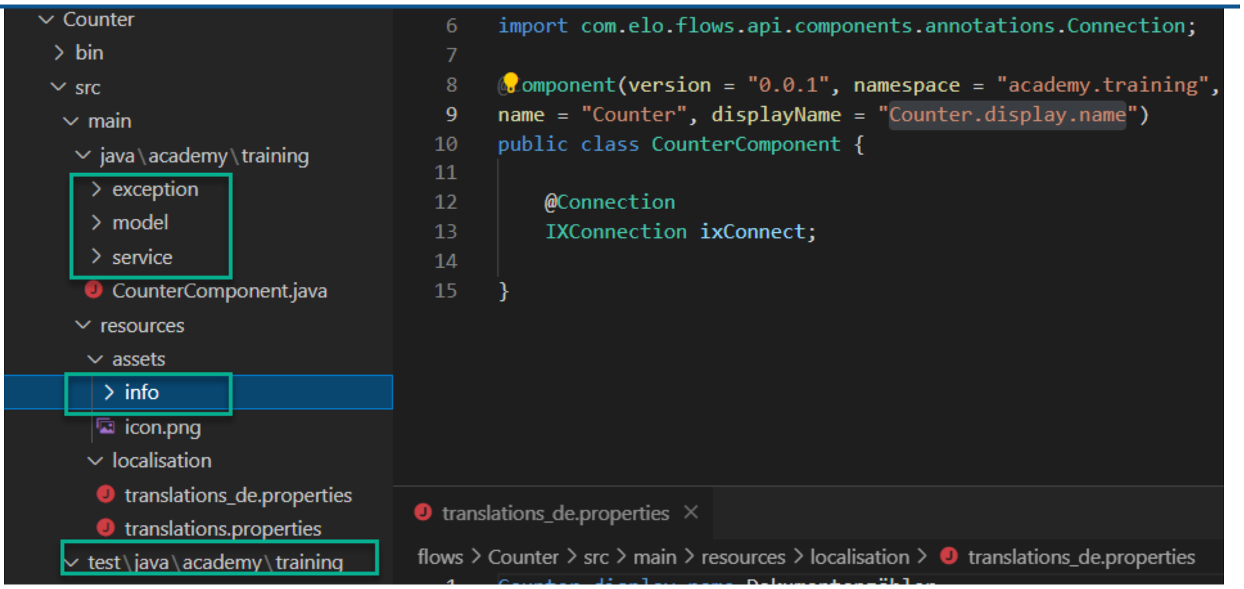
```
Counter.display.name=Document Counter
```

4. Enter the following in the localization files in the *localization* folder:

   `translations\_de.properties` and `translations.properties`

   Key: `Counter.display.name`

   Values: `Dokumentenzähler` and `Document Counter`.



```
import com.elo.flows.api.components.annotations.Connection;

@Component(version = "0.0.1", namespace = "academy.training",
name = "Counter", displayName = "Counter.display.name")
public class CounterComponent {

    @Connection
    IXConnection ixConnect;

}
```

Counter
bin
src
main
java\academy\training
exception
model
service
CounterComponent.java
resources
assets
info
icon.png
localisation
translations_de.properties
translations.properties
test\java\academy\training
translations_de.properties
flows > Counter > src > main > resources > localisation > translations_de.properties

5. Add additional levels to the project structure.

You will find an example implementation of this requirement in the Appendix.

ELOas module

In this task, we will implement the function from the ELOas module (to a certain extent).

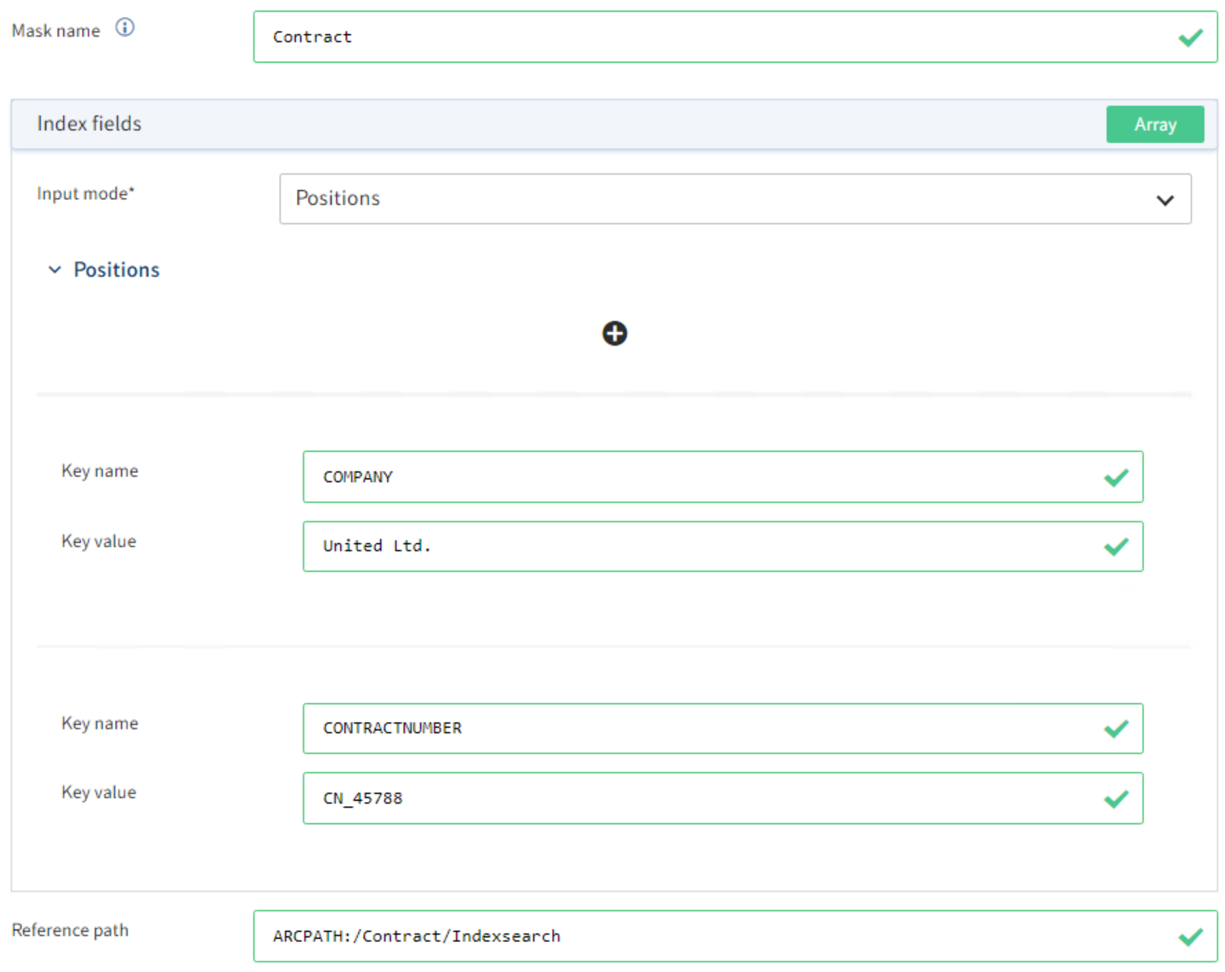
We want to trigger a search (index search) that defines the metadata form and group fields as search parameters.

The objects found are referenced in a specified repository structure.

Requirements (general)

- Specify a metadata form
- Select a metadata form from the list of suggestions
- Enter multiple fields, e.g. name and search value
- Specify the filing path for references to the objects found
- Provide a trigger (REST)

Configuration interface on the *Settings* tab.



Mask name	Contract

Index fields

Input mode*	Positions

Positions

Key name	COMPANY
Key value	United Ltd.
Key name	CONTRACTNUMBER
Key value	CN_45788

Reference path	ARCPATH:/Contract/Indexsearch

Participant task (optional)

This task is only done if participants have made suggestions for implementing very specific requirements.