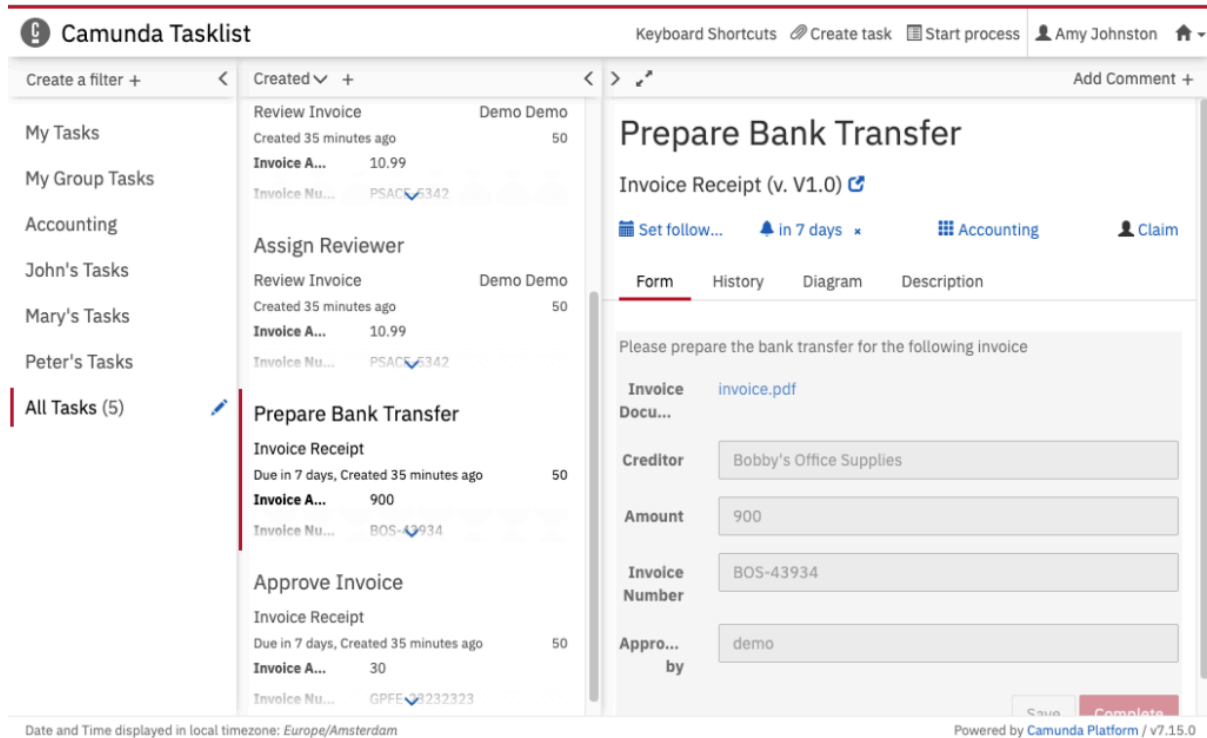


Mit einer Workflow-Engine arbeiten

Umsetzung eines Prozesses in Camunda



The screenshot displays the Camunda Tasklist interface. On the left, a sidebar lists various task categories: 'My Tasks', 'My Group Tasks', 'Accounting', 'John's Tasks', 'Mary's Tasks', 'Peter's Tasks', and 'All Tasks (5)'. The main area shows a list of tasks, including 'Review Invoice', 'Assign Reviewer', 'Prepare Bank Transfer', and 'Approve Invoice'. The 'Prepare Bank Transfer' task is selected, and its details are shown on the right. The task is titled 'Prepare Bank Transfer' and is an 'Invoice Receipt (v. V1.0)'. It is due in 7 days and has a value of 900. The task is assigned to 'demo' and is part of the 'Accounting' group. The task details form includes fields for 'Invoice Docu...' (invoice.pdf), 'Creditor' (Bobby's Office Supplies), 'Amount' (900), 'Invoice Number' (BOS-43934), and 'Appro... by' (demo). The interface also shows a 'Set follow...' button and a 'Claim' button. The bottom of the interface indicates the date and time displayed in local timezone: Europe/Amsterdam, and the version of the Camunda Platform: v7.15.0.

Mit Camunda lassen sich Geschäftsprozesse aus einem BPMN-Modell direkt in einen automatisierbaren Ablauf übertragen. Dieses Dokument gibt einen Einstieg in die Arbeit mit dieser Workflow-Engine.

Inhaltsverzeichnis

Einleitung	3
Was wird benötigt?	3
Vorgehen grob	3
Schritt-für-Schritt-Anleitung	3
Basis-Modell entwerfen und deployen	4
Erweiterung um Aufgabentypen	10
Formular definieren	13
Einem UserTask ein Formular zuordnen	16
Erweiterung um ein exklusives Gateway	19
Resultierende Dateien	21
BPMN	21
Form	24

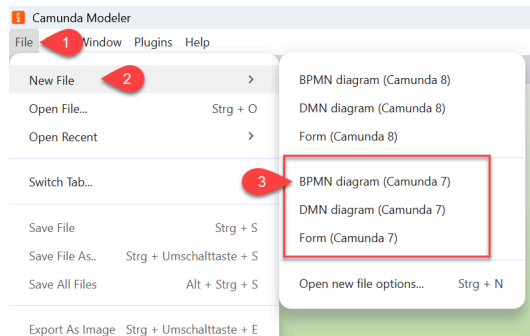
Einleitung

Camunda ist ein Beispiel eines Process Engines, oder Workflow Engine. In diesem Dokument wird gezeigt, wie ein einfacher Prozess in Camunda nachgebildet werden kann.

Was wird benötigt?

- Zur Modellierung mit BPMN benötigen Sie die Desktop-Version des Camunda Modelers (⇒ [Download](#)).
- Zum Umsetzen des Geschäftsprozess benötigen Sie die Camunda Engine (⇒ [Installation](#)).

Sehr wichtig: Wir arbeiten mit der Camunda Version 7. Verwenden Sie für alle folgenden Arbeiten im Camunda Modeler die Vorlagen zur Version 7:



Vorgehen grob

Das Vorgehen erfolgt grob in drei Schritten:

- 1.) Zuerst wird der Prozess mit BPMN modelliert.
- 2.) Danach wird das Modell in die Camunda Engine übertragen (deployed).
- 3.) In einer von Camunda gelieferten Webapp wird der Prozess gestartet (= durchgeführt)

Die Engine arbeitet danach den modellierten Prozess ab und führt die nötigen Tasks aus.

Schritt-für-Schritt-Anleitung

In den folgenden Kapiteln wird mit BPMN ein Modell erstellt. Nach der Übergabe an die Workflow-Engine von Camunda lässt sich der Prozess bereits starten.

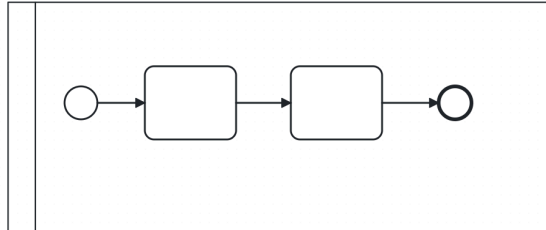
Um den Ablauf benutzerfreundlicher zu machen, definieren Sie ein Formular.

Schliesslich wird der Ablauf noch um eine Verzweigung (Gateway) ergänzt.

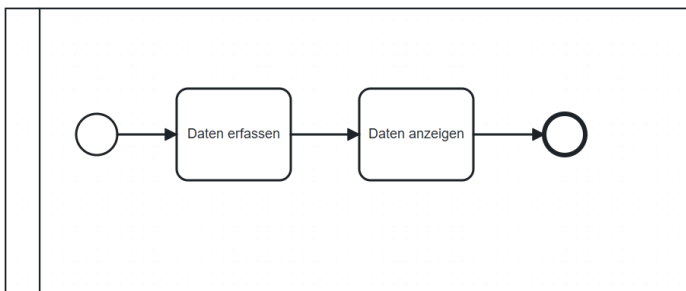
Basis-Modell entwerfen und deployen

Im Folgenden erstellen wir im Programm "Camunda Modeler" eine BPMN-Datei.

Erstellen Sie im Programm "Camunda Modeler" einen Prozess, bestehend aus einem Pool, einem Startevent, zwei Tasks und einem Endevent.

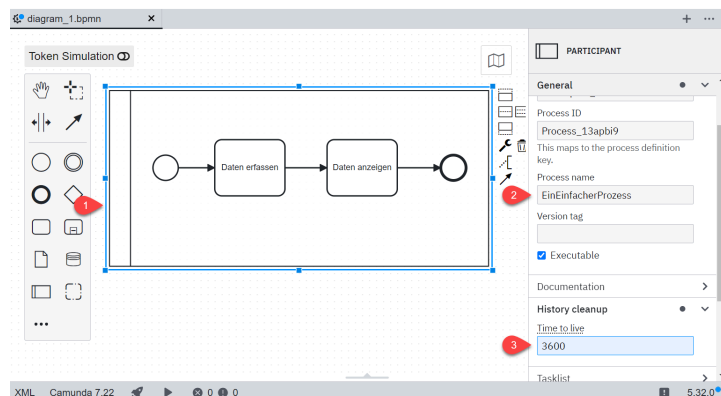


Benennen Sie die Tasks



Markieren Sie den Pool (1) und vergeben Sie einen Namen (2) für den Prozess.
Erfassen Sie unter Register "History cleanup" bei "Time to live" (3) die Angabe 3600.

Hinweis: Das Fenster rechts (Eigenschaftsfenster) kann mit Ctrl-P eingeblendet werden.

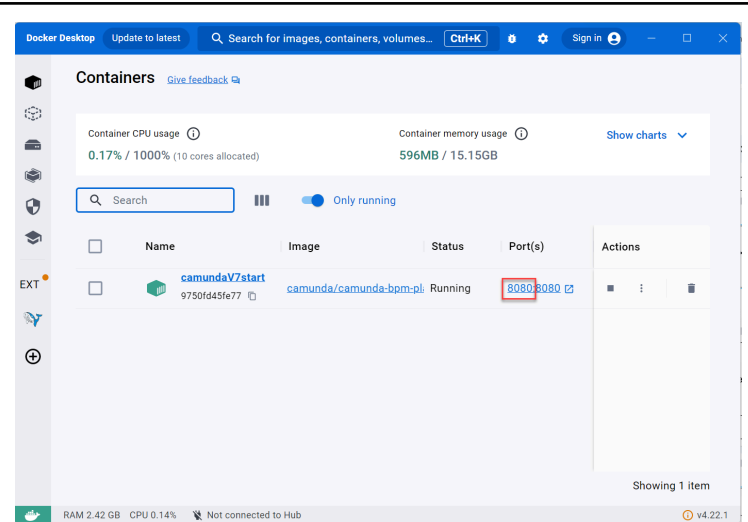


Speichern Sie die Datei!

Die wichtigsten Dinge für einen Prozess sind damit erfasst. Der modellierte Prozess kann nun an die Workflow Engine übergeben werden.

Stellen Sie sicher, dass die Camunda Engine gestartet ist und der Port 8080 auf einen Host-Port gemappt ist (im Beispiel ebenfalls 8080).

Falls Sie mit dem von der TBZ bereitgestellten Linux-Server arbeiten, schauen Sie, dass Sie den Tunnel in Wireguard aktiviert haben!

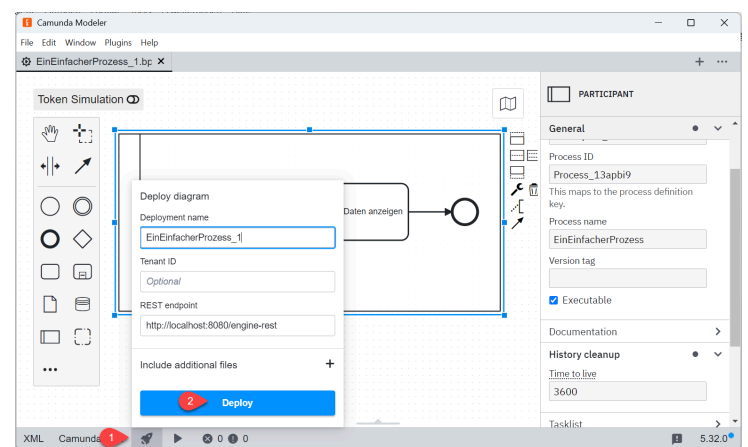


Klicken Sie auf das Icon mit der Rakete am unteren Rand (1).

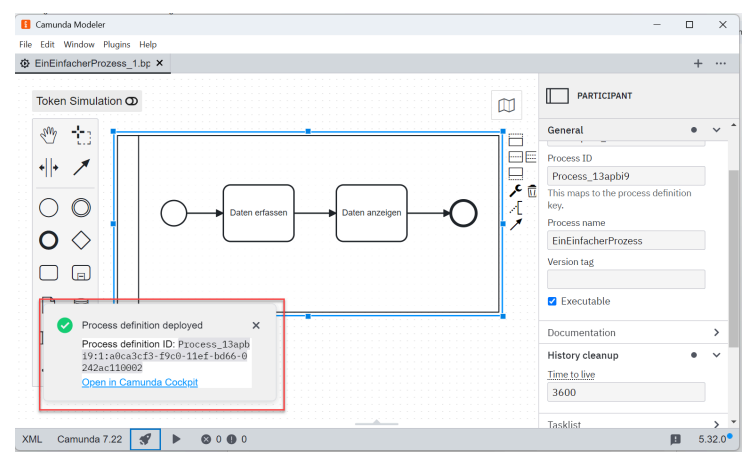
Hinweis:

Der Deploy-Button ist disabled, falls die Camunda Engine nicht gestartet ist oder nicht auf dem Standardport 8080 erreicht werden kann.

Falls Sie mit dem TBZ-Server arbeiten, verwenden Sie statt "localhost" die IP-Adresse des Servers.



Falls alles gut gelaufen ist, erhalten Sie die Meldung "Process definition deployed".



Falls Sie eine Fehlermeldung erhalten, drücken Sie auf den darunterstehenden Link.

Hinweis

Lesen Sie die Fehlermeldung genau! Sie gibt Ihnen meist sehr detaillierte Angaben über den Fehler.

Falls eine Zeile erwähnt ist, drücken Sie auf "XML" (unten links), um sich die Datei im XML-Format anzeigen zu lassen.

Deployment failed
[See the log for further details.](#)

XML Camunda 7.22

Camunda Modeler
File Edit Window Plugins Help
EinEinfacherProzess_1.bpm

Token Simulation

General
ID
Collaboration_1139710
Documentation
Extension properties

Output Problems

ENGINE-09905 Could not parse BPMN process. Errors:
* ENGINE-12019 History Time To Live (TTL) cannot be null. TTL is necessary for the History Cleanup to work. The following options are possible:
* Set historyTimeToLive in the model
* Set a default historyTimeToLive as a global process engine configuration
* (Not recommended) Deactivate the enforceTTL config to disable this check: ENGINE-12018 History Time To Live (TTL) cannot be null. TTL is necessary for the History Cleanup to work. The following options are possible:
* Set historyTimeToLive in the model
* Set a default historyTimeToLive as a global process engine configuration

XML Camunda 7.22

Der Prozess ist nun der Camunda Engine bekannt.

Wir werden jetzt über die Weboberfläche von Camunda den Prozess starten.

Rufen Sie die Adresse "http://localhost:8080" in einem Browser auf.

Sie erhalten den Login-Screen für die Webapp von Camunda.

Loggen Sie sich ein mit
Username: demo
Password: demo

Hinweis

Das Konto demo/demo wurde als Superuser definiert. Mit diesem Konto sehen Sie alles und haben alle Berechtigungen.

Camunda Welcome

localhost:8080/camunda/app/welcome/default/#/login

Heads-up – it is time to try out Camunda 8. The last release of Camunda 7 CE will be in October 2025. If you have any more questions, please get in touch with us!

CAMUNDA
Welcome

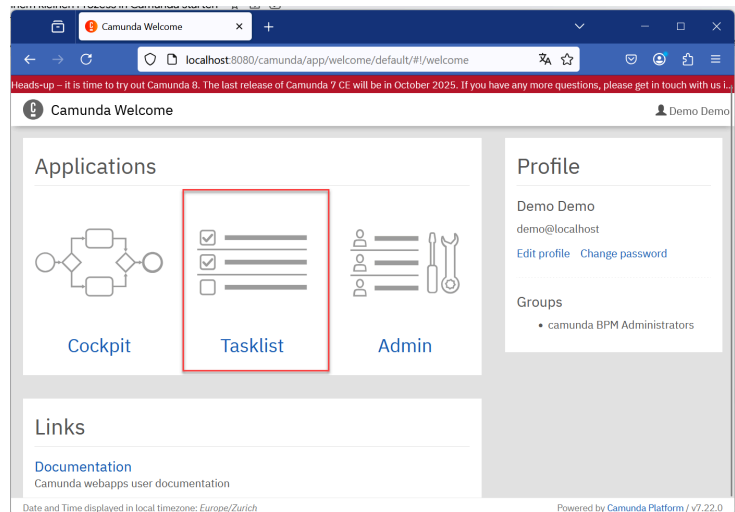
Username
Password
Log in

Date and Time displayed in local timezone: Europe/Zurich
Powered by Camunda Platform / v7.22.0

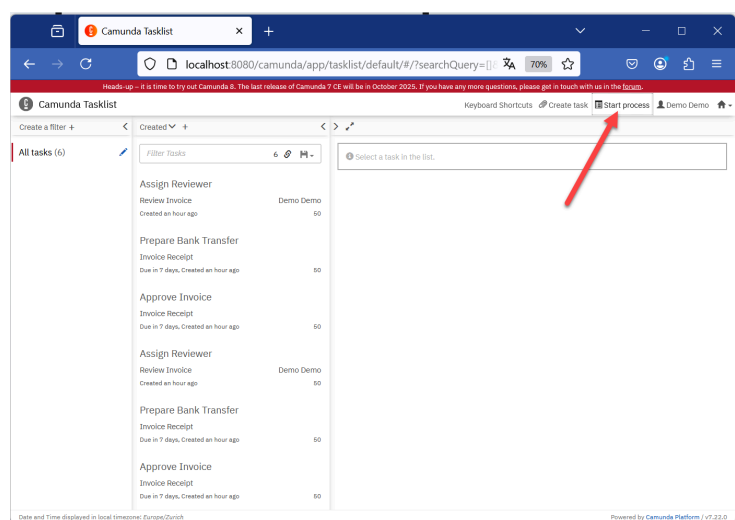
Die Camunda-Oberfläche unterscheidet drei Bereiche:

- Cockpit
- Tasklist
- Admin

Für die tägliche Arbeit mit Prozessen wurde die "Tasklist" geschaffen. Klicken Sie darauf!



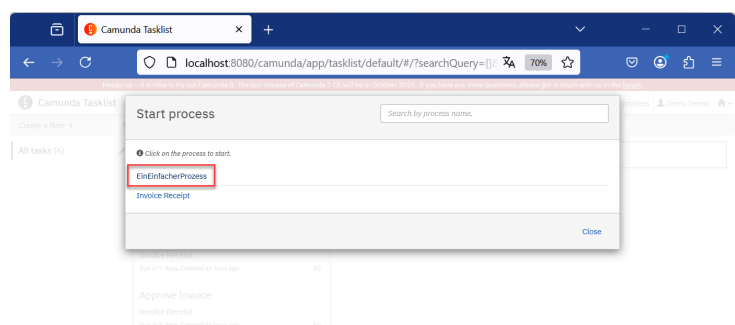
Drücken sie auf "Start process".



Sie erhalten eine Liste mit den verfügbaren Prozessen.

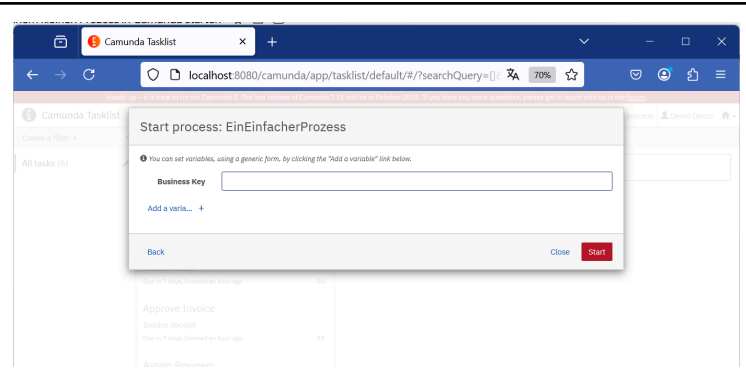
Falls Sie alles richtig gemacht haben, sollten Sie hier einen Eintrag mit dem Namen finden, welchem Sie Ihrem Prozess vergeben haben.

Klicken Sie auf den Eintrag.

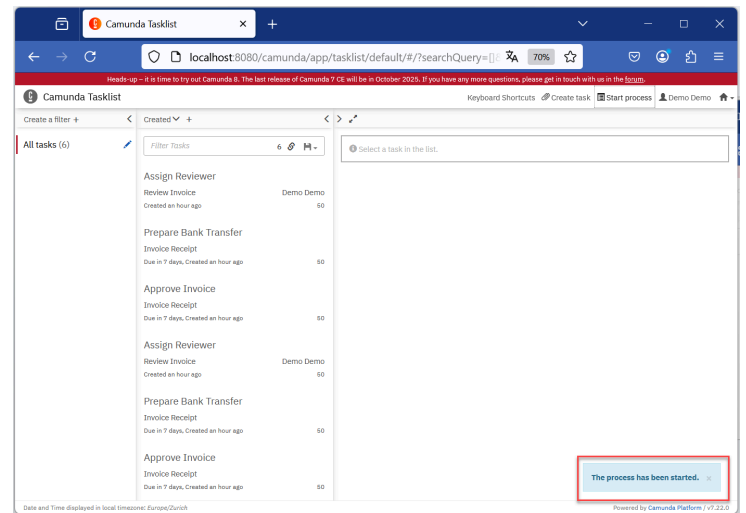


Sie erhalten ein Fenster, in welcher Sie dem Prozess beim Start Angaben mitgeben könnten.

Drücken Sie auf "Start"!



Die Engine meldet, dass der Prozess gestartet wurde.



Wie Sie sehen, kennt Camunda Ihren Prozess und Sie konnten ihn erfolgreich starten. Das ist aber auch schon alles, was Sie sehen. Die Engine hat den Prozess danach sehr schnell beendet.

Was ist passiert?

Nach dem Startevent geht die Engine dem Sequenzfluss nach.

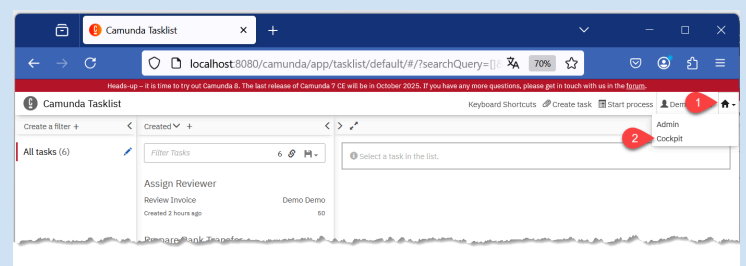
Sie findet einen ersten Task "Eingabe erfassen". Diesem wurde aber kein Aufgabentyp zugeordnet. Das heisst für die Engine "Es gibt nichts zu tun. Einfach weiter dem Sequenzfluss nach!"

Und nach dem zweiten Task "Daten ausgeben" - ebenfalls ohne Angabe - landet Sie auf dem Endevent und beendet die Prozessinstanz. Schluss!

Hinweis

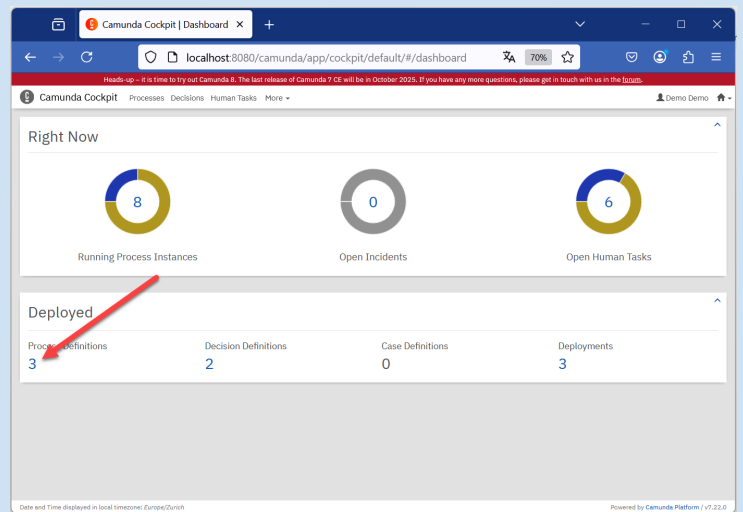
Im Cockpit können Sie Angaben zu den vorhandenen Prozessdefinition sehen.

Klicken Sie dazu auf das Haus-Icon (1) und danach auf "Cockpit" (2).

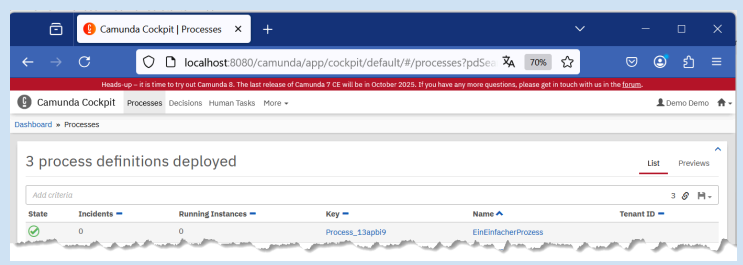


Sie sehen die Anzahl der Prozesse, welche Camunda bekannt sind (siehe Pfeil).

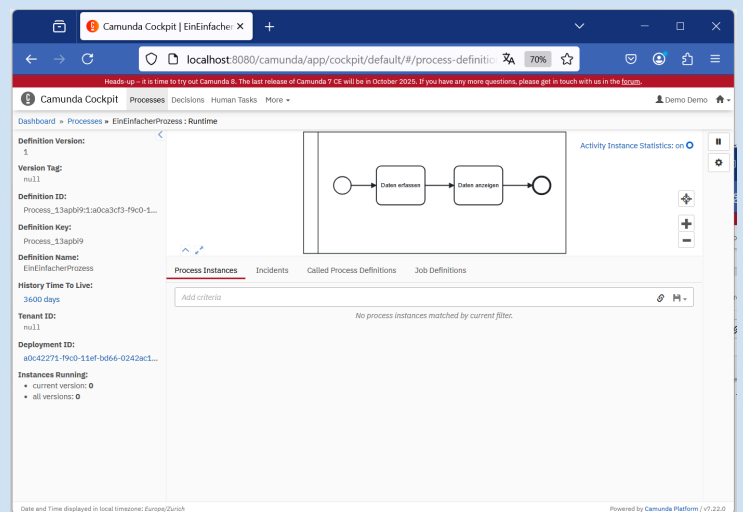
Für nähere Angaben zu diesen Prozessen klicken Sie auf die Zahl.



Klicken Sie auf den Prozess...



... erhalten Sie Angaben zu diesem Prozess.

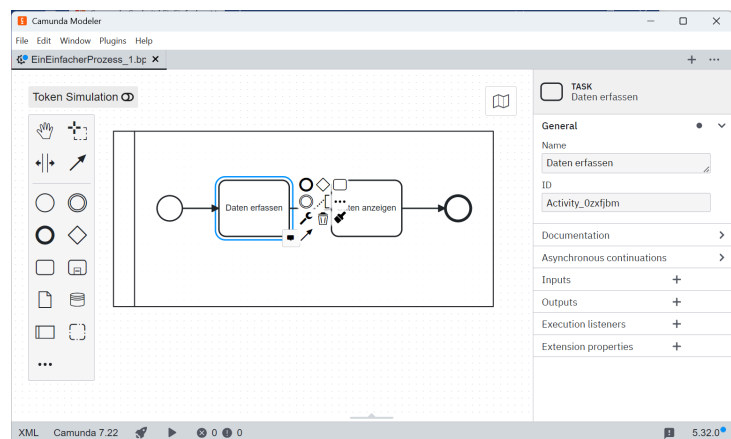


Erweiterung um Aufgabentypen

Wenn wir Tasks einen Aufgabentyp zuweisen, geben wir der Engine eine Anweisung, was Sie zu tun hat. Je nach Typ der Aufgabe müssen oder können wir weitere Details regeln.

Im Folgenden werden wir den beiden Tasks den Aufgabentyp “User Task” zuordnen.

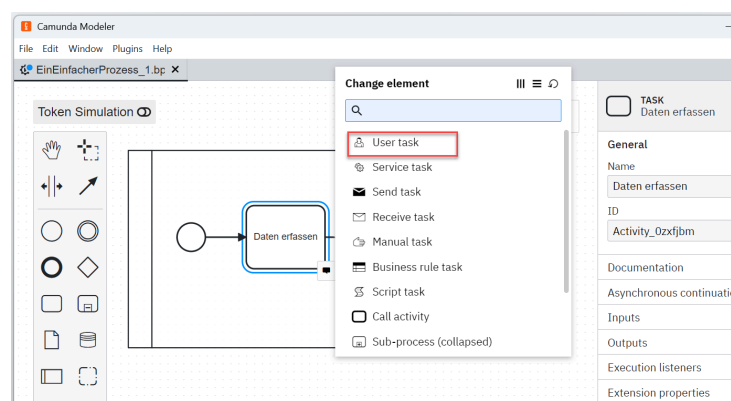
Öffnen Sie die BPMN-Datei und markieren Sie den Task “Daten erfassen”.



und wählen Sie über den Schraubenschlüssel...



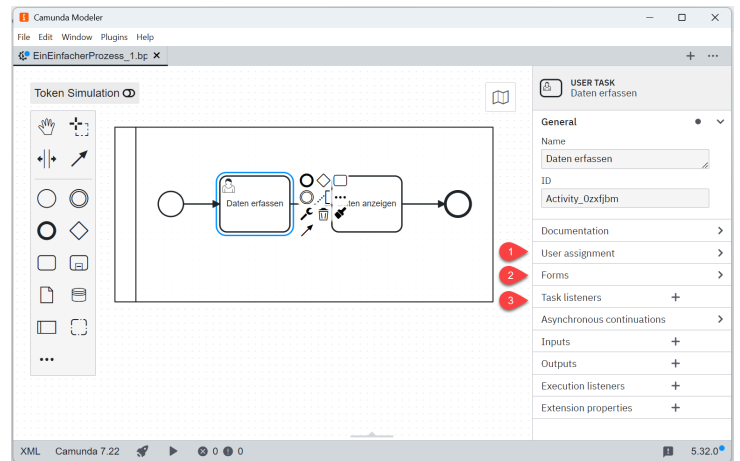
den Typ “User task” aus.



Haben Sie gesehen, dass nun drei neue Bereiche im Eigenschaftsfenster erschienen sind?

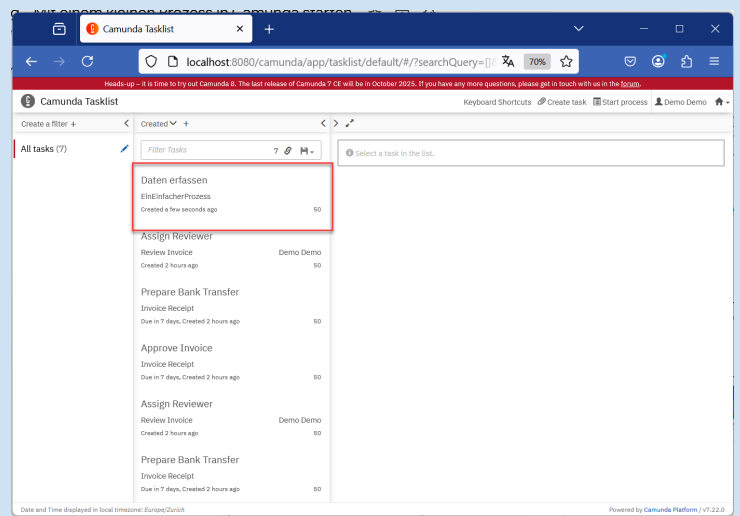
Bei einem Benutzertask (User Task) können Sie bspw. angeben, wer diesen zu bearbeiten hat (1) und welches Formular verwendet werden soll (2).

Bei User Tasks kann auch angegeben werden, welche Skripts oder Programme vor oder nach dem Erledigen aufgerufen werden sollen (3).



Wenn Sie möchten, können Sie nach dieser Änderung das Modell über das Raketesymbol deployen.

Nach dem Starten des Prozesses (und nach einem Refresh) werden Sie in der Tasklist einen Task "Daten erfassen" finden.



Wenn Sie diesen Task anklicken, erhalten Sie auf der rechten Seite die Details zum Task.

Hinweis

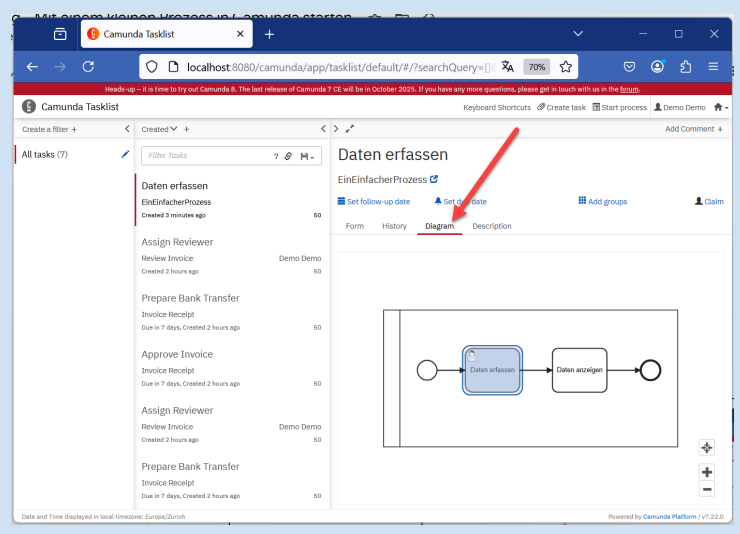
*Sie merken: Es ist alles disabled, keine Eingabe ist möglich.
Der Grund: Der Task ist noch keiner Person zugeordnet.
Wenn Sie ihn übernehmen möchten, müssen Sie ihn "claimen" (siehe Pfeil).*

Danach könnten Sie beliebig viele Variablen erfassen.

The screenshot shows the Camunda Tasklist interface. On the left, a list of tasks is displayed, including 'Daten erfassen', 'Assign Reviewer', 'Review Invoice', 'Prepare Bank Transfer', 'Approve Invoice', and 'Assign Reviewer'. The 'Daten erfassen' task is selected, and its details are shown on the right. The details panel includes tabs for 'Form', 'History', 'Diagram', and 'Description'. The 'Form' tab is active, showing a form titled 'EinEinfacherProzess'. The form has fields for 'Set follow-up date', 'Set due date', 'Add groups', and 'Claim'. A red arrow points to the 'Claim' button in the top right corner of the task details panel.

The screenshot shows the 'Daten erfassen' form in the Camunda Tasklist interface. The form is titled 'EinEinfacherProzess' and has tabs for 'Form', 'History', 'Diagram', and 'Description'. The 'Form' tab is active, showing a form with fields for 'Name', 'Type', and 'Value'. The 'Name' field is labeled 'Vornname' and the 'Value' field is labeled 'Peter'. The 'Type' field is labeled 'String'. The 'Name' field is also labeled 'Nachname' and the 'Value' field is labeled 'Muster'. The 'Form' tab is active, showing a form with fields for 'Name', 'Type', and 'Value'. The 'Name' field is labeled 'Vornname' and the 'Value' field is labeled 'Peter'. The 'Type' field is labeled 'String'. The 'Name' field is also labeled 'Nachname' and the 'Value' field is labeled 'Muster'.

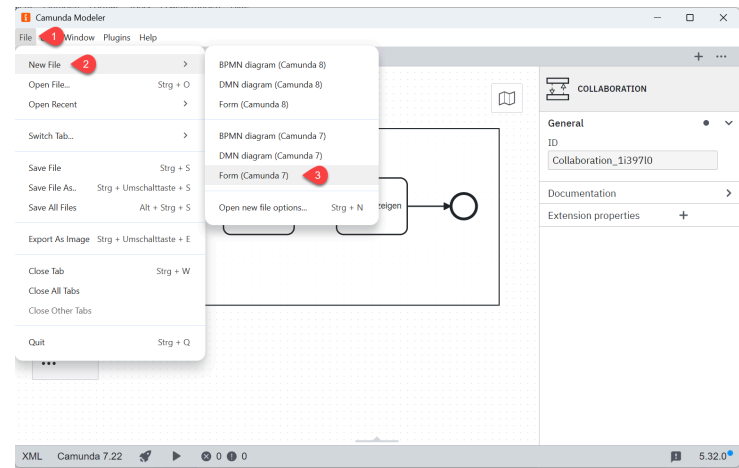
Über die Registerlasche "Diagram" können Sie feststellen, mit welchem Task innerhalb des Prozesses Sie sich gerade beschäftigen.



Formular definieren

Damit die Benutzer auch wissen, welche Daten Sie zu erfassen haben, können wir bei "User Tasks" ein Formular vorgeben.

Erstellen Sie ein neues Formular vom Typ "Form (Camunda 7)"

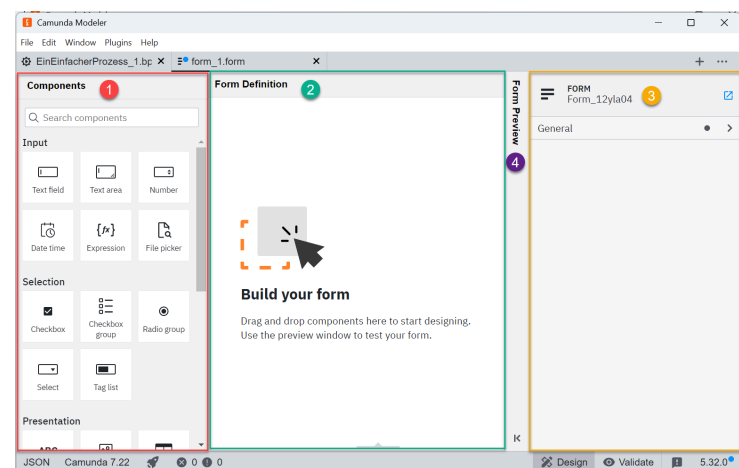


Für die Definition des Formulars steht Ihnen links (1) eine Liste mit Komponenten (Controls) zur Verfügung.

In der Mitte (2) sehen Sie die Definition des Formulars.

Rechts (3) sind Eigenschaften des aktuell markierten Controls zu sehen.

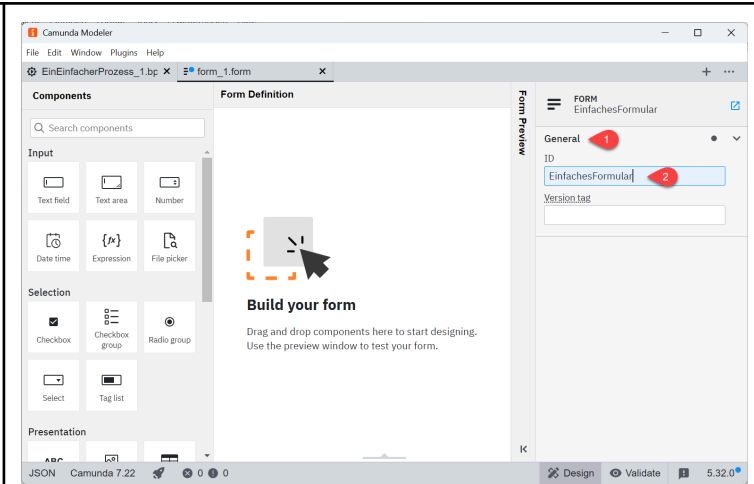
Es gibt noch einen Bereich (4), welcher das Formular so anzeigt, wie es der Benutzer sehen würde (Preview).



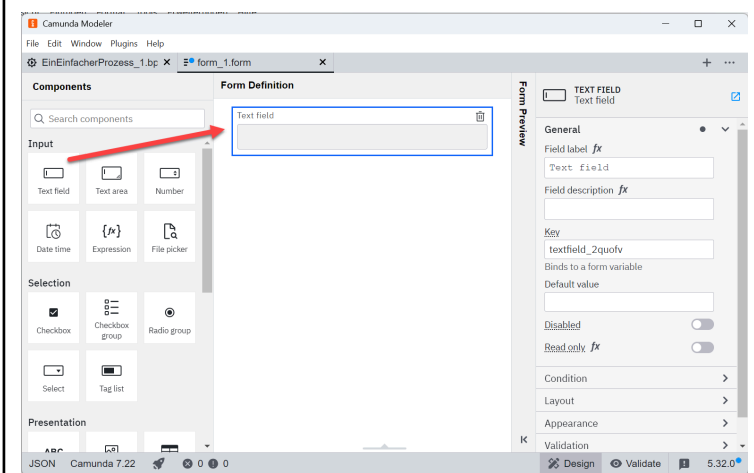
Geben Sie dem Formular im Bereich "General" (1) eine ID (2).

Hinweis

Über diese ID findet die Engine später das zu einem UserTask gehörende Formular. Diese ID werden wir später im BPMN beim UserTask angeben müssen.

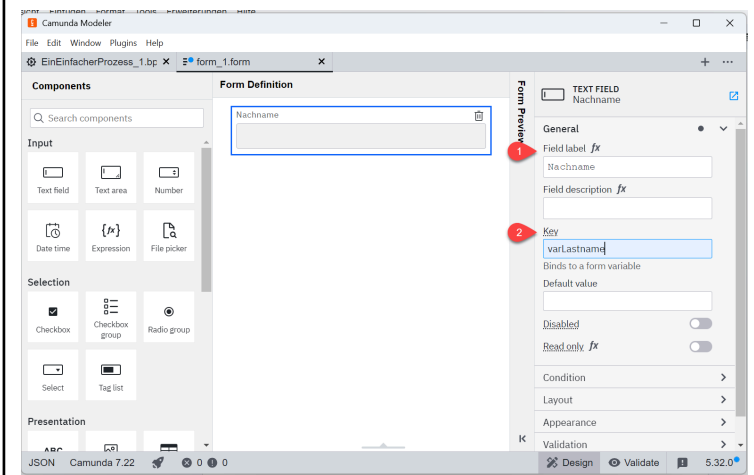


Ziehen Sie aus der Components-Liste ein "Text field" in den Bereich "Form Definition".

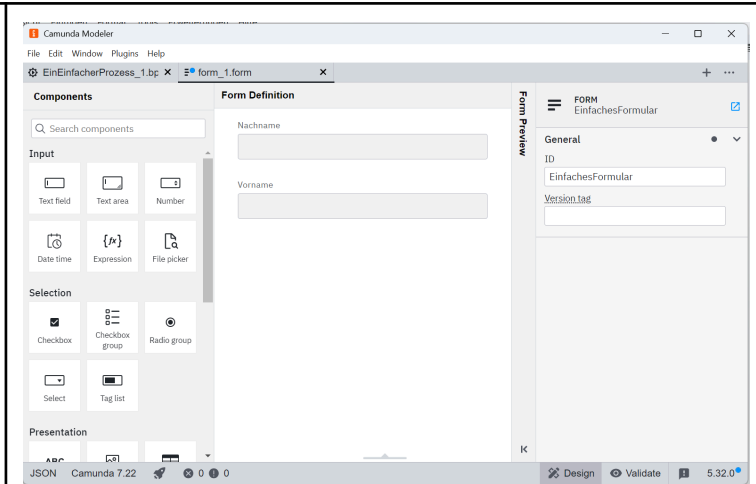


Geben Sie beim "Field label" (1) den Text an, welcher dem Benutzer erklärt, was er im folgenden Control zu erfassen hat.

Erfassen Sie bei "Key" (2) einen Namen für die Variable, welche den Inhalt des Controls übernehmen soll.



Erweitern Sie das Formular allenfalls noch um zusätzliche Controls und speichern Sie danach das Formular!



Kopieren Sie die ID des Formulars in die Zwischenablage.
Öffnen Sie dann wieder die BPMN-Datei. Sie werden die ID dort benötigen.

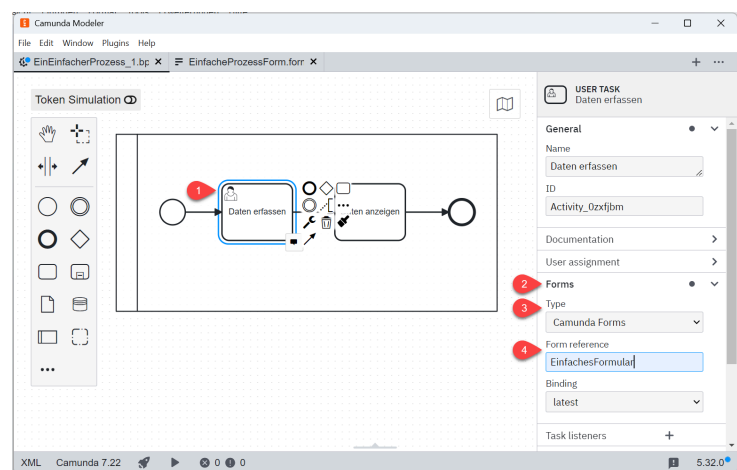
Einem UserTask ein Formular zuordnen

Sie haben ein Formular erstellt und diesem eine ID vergeben. Im folgenden werden Sie aus dem BPMN heraus dieses Formular referenzieren.

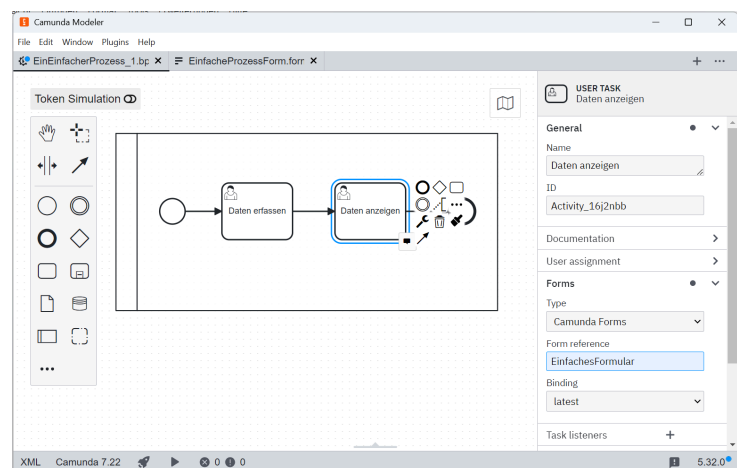
Markieren Sie den UserTask, welchem Sie ein Formular zuordnen möchten (1).

Im Bereich "Forms" (2) wählen Sie den Typ (3) "Camunda Forms" aus.

Unter "Forms reference" (4) geben Sie die ID des Formulars an (diese sollten Sie ja noch in der Zwischenablage haben).



Sie können für den zweiten Task dasselbe Formular verwenden und dieselben Eigenschaften definieren.



Wenn Sie nun den Prozess erneut deployen und starten, werden Sie sehen, dass Sie auf einen Fehler stossen:

"Form failure:"

Frage

Haben Sie eine Idee, warum dieser Fehler erscheint?

Daten erfassen

EinEinfacherProzess

Set follow-up date

Set due date

Add groups

Claim

Form History Diagram Description

Form failure:

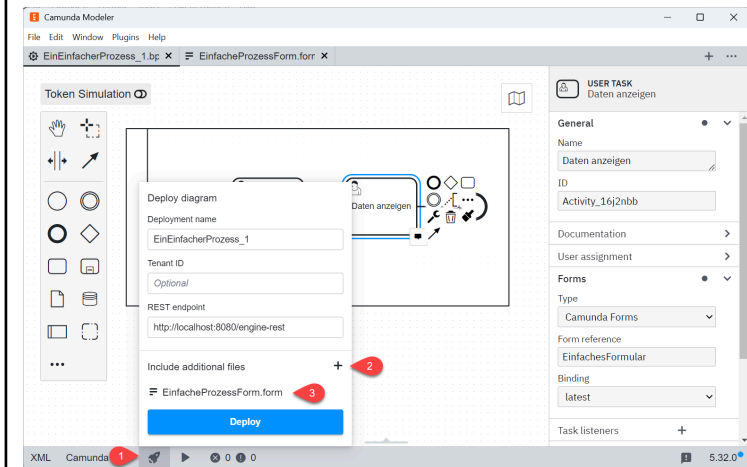
Save Complete

Antwort

Wenn wir im BPMN ein Formular referenzieren, muss der Engine natürlich auch die

Definition dieses Formulars übergeben werden.

Beim Deployen (1) müssen wir der Engine nebst dem BPMN über “include additional files” (2) das Formular (3) mitgeben.

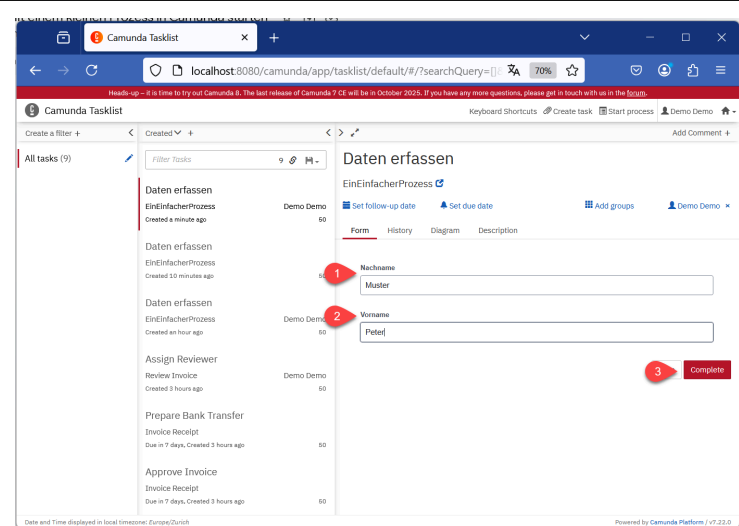


Nun sollte der Prozess korrekt durchlaufen und bei den Usertasks das Formular angezeigt werden.

Mit “Claim” übernehmen Sie den Task. Der Task ist dann “Demo” zugeordnet.

Sie können danach Daten erfassen (1, 2).

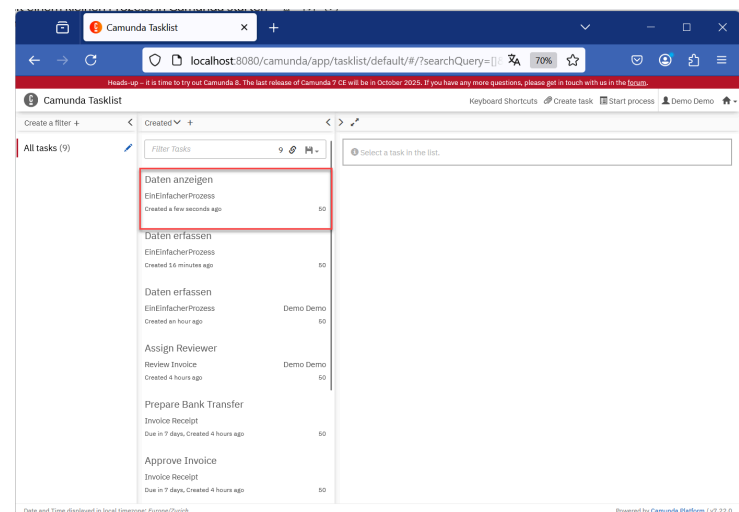
Durch Drücken auf “Complete” (3) beenden Sie den Task.



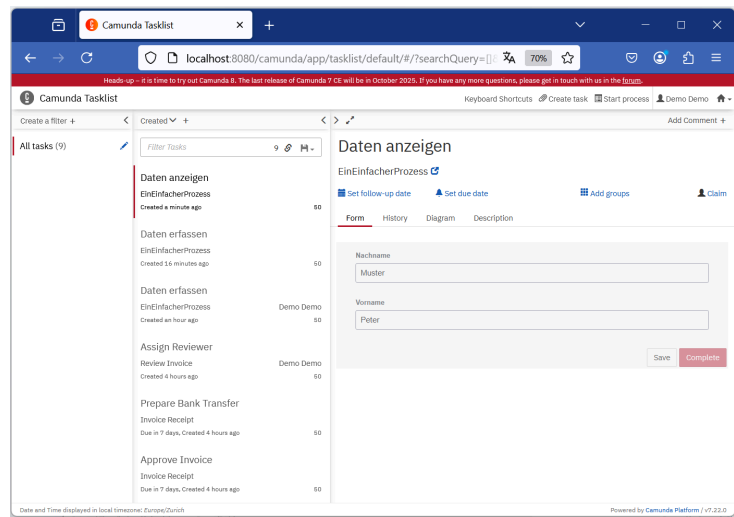
Die Engine hat anhand des Sequenzflusses festgestellt, dass der Usertask “Daten anzeigen” als nächstes auszuführen ist.

Da dies ebenfalls ein Usertask ist, reiht sie diesen in die Taskleiste ein.

Aus diesem Grund sollten Sie nun in der Liste der Tasks den Task “Daten ausgeben” sehen.



Beim Anwählen dieses Tasks sehen Sie, dass die Informationen, welche Sie im vorherigen Task erfasst haben, weitergegeben wurden.



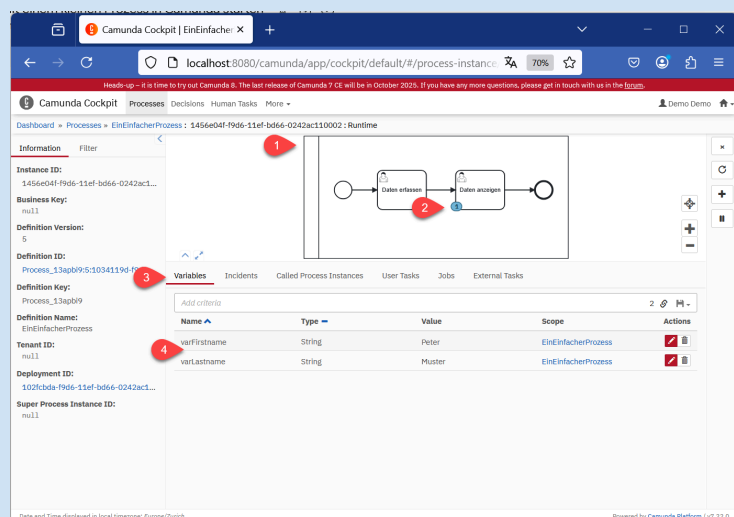
Bei jedem Start eines Prozesses wird eine **Prozessinstanz** erstellt.

Im Cockpit können Sie sehen, wie weit der Prozess einer bestimmten Prozessinstanz fortgeschritten ist.

Wählen Sie dazu im Cockpit die entsprechende Prozessdefinition und Prozessinstanz aus.

Im Diagramm (1) sehen Sie, wo sich der Token der Prozessinstanz gerade befindet (2).

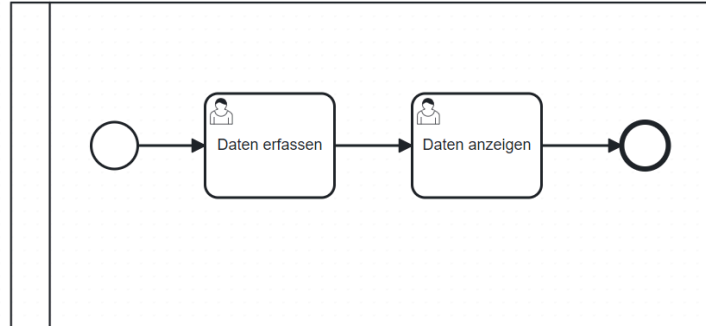
In der Lasche "Variables" (3) erhalten Sie eine Übersicht über die Variablen mit der Möglichkeit, diese zu ändern.



Erweiterung um ein exklusives Gateway

Bis jetzt haben wir einen einfachen sequentiellen Ablauf implementiert.

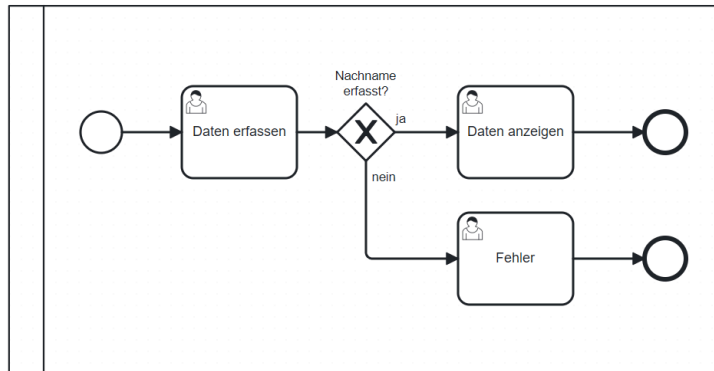
Im Folgenden soll dieser Ablauf um eine Verzweigung erweitert werden.



Erweitern Sie das BPMN gemäss nebenstehendem Diagramm.

Hinweis

Bis hierher müssen noch keine Angaben im Eigenschaftsfenster definiert werden.



Damit die Engine richtig arbeiten kann, müssen wir noch einige Eigenschaften auf den Pfaden setzen.

Markieren Sie den "nein"-Pfad (1) und definieren Sie im Bereich "Condition" (2) eine Bedingung vom Typ "Expression" (3).

Die Bedingung muss wie folgt definiert werden:

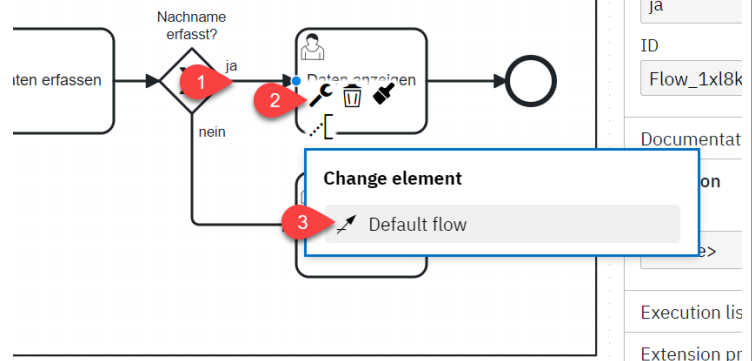
```
${ varLastname == "" }
```

Auf dem "ja"-Pfad könnten wir ebenfalls eine Bedingung definieren

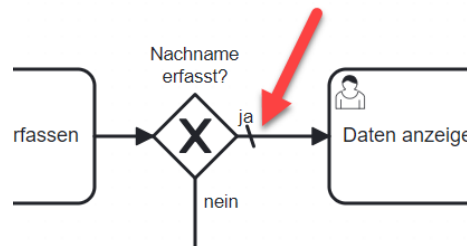
```
${ varLastname != "" }
```

Wir machen uns die Sache aber einfach und erklären den Pfad als "Defaultpfad". Er soll verwendet werden, wenn die Bedingungen der anderen Pfade nicht zutreffen.

Dazu den Pfad markieren (1), den Schraubenschlüssel anwählen (2) und "Default flow" (3) wählen.



Defaultflow als Strich sichtbar:

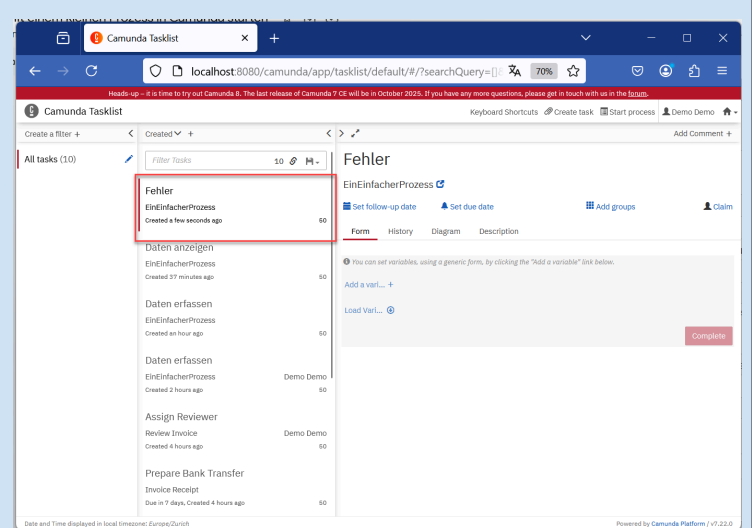


Speichern Sie die Datei und übertragen Sie diese an die Engine.

Test

Wenn Sie im ersten Usertask "Daten erfassen" keinen Nachnamen eintragen, sollten Sie in der Taskleiste den UserTask "Fehler" sehen.

Andernfalls sollten Sie wie vorher den Task "Daten anzeigen" sehen.



Resultierende Dateien

Als Resultat der vorherigen Anleitung sind zwei Dateien entstanden. Die Inhalte der beiden Dateien finden Sie nachfolgend.

BPMN

```
<?xml version="1.0" encoding="UTF-8"?>
<bpmn:definitions xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
xmlns:camunda="http://camunda.org/schema/1.0/bpmn"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
xmlns:modeler="http://camunda.org/schema/modeler/1.0" id="Definitions_007psl5"
targetNamespace="http://bpmn.io/schema/bpmn" exporter="Camunda Modeler"
exporterVersion="5.32.0" modeler:executionPlatform="Camunda Platform"
modeler:executionPlatformVersion="7.22.0">
  <bpmn:collaboration id="Collaboration_1i397l0">
    <bpmn:participant id="Participant_03363un" processRef="Process_13apbi9" />
  </bpmn:collaboration>
  <bpmn:process id="Process_13apbi9" name="EinEinfacherProzess" isExecutable="true"
camunda:historyTimeToLive="3600">
    <bpmn:startEvent id="StartEvent_1">
      <bpmn:outgoing>Flow_008de66</bpmn:outgoing>
    </bpmn:startEvent>
    <bpmn:sequenceFlow id="Flow_008de66" sourceRef="StartEvent_1"
targetRef="Activity_0zxfjbm" />
    <bpmn:sequenceFlow id="Flow_06boo6n" sourceRef="Activity_0zxfjbm"
targetRef="Gateway_093kenq" />
    <bpmn:sequenceFlow id="Flow_1pl304m" sourceRef="Activity_16j2nbb"
targetRef="Event_14ehgqs" />
    <bpmn:userTask id="Activity_0zxfjbm" name="Daten erfassen"
camunda:formRef="EinfachesFormular" camunda:formRefBinding="latest">
      <bpmn:extensionElements />
      <bpmn:incoming>Flow_008de66</bpmn:incoming>
      <bpmn:outgoing>Flow_06boo6n</bpmn:outgoing>
    </bpmn:userTask>
    <bpmn:exclusiveGateway id="Gateway_093kenq" name="Nachname erfasst?"
default="Flow_1xl8kpo">
      <bpmn:incoming>Flow_06boo6n</bpmn:incoming>
      <bpmn:outgoing>Flow_1xl8kpo</bpmn:outgoing>
      <bpmn:outgoing>Flow_00j6ei0</bpmn:outgoing>
    </bpmn:exclusiveGateway>
```

```
<bpmn:sequenceFlow id="Flow_1xl8kpo" name="ja" sourceRef="Gateway_093kenq"
targetRef="Activity_16j2nbb" />
<bpmn:sequenceFlow id="Flow_00j6ei0" name="nein" sourceRef="Gateway_093kenq"
targetRef="Activity_1mtzecx">
  <bpmn:conditionExpression xsi:type="bpmn:tFormalExpression">${ varLastname ==
""}</bpmn:conditionExpression>
</bpmn:sequenceFlow>
<bpmn:sequenceFlow id="Flow_1pdi1wf" sourceRef="Activity_1mtzecx"
targetRef="Event_1e2p0uz" />
<bpmn:endEvent id="Event_14ehgqs">
  <bpmn:incoming>Flow_1pl304m</bpmn:incoming>
</bpmn:endEvent>
<bpmn:userTask id="Activity_16j2nbb" name="Daten anzeigen"
camunda:formRef="EinfachesFormular" camunda:formRefBinding="latest">
  <bpmn:incoming>Flow_1xl8kpo</bpmn:incoming>
  <bpmn:outgoing>Flow_1pl304m</bpmn:outgoing>
</bpmn:userTask>
<bpmn:endEvent id="Event_1e2p0uz">
  <bpmn:incoming>Flow_1pdi1wf</bpmn:incoming>
</bpmn:endEvent>
<bpmn:userTask id="Activity_1mtzecx" name="Fehler">
  <bpmn:incoming>Flow_00j6ei0</bpmn:incoming>
  <bpmn:outgoing>Flow_1pdi1wf</bpmn:outgoing>
</bpmn:userTask>
</bpmn:process>
<bpmndi:BPMNDiagram id="BPMNDiagram_1">
  <bpmndi:BPMNPlane id="BPMNPlane_1" bpmnElement="Collaboration_1i397l0">
    <bpmndi:BPMNShape id="Participant_03363un_di"
bpmnElement="Participant_03363un" isHorizontal="true">
      <dc:Bounds x="120" y="70" width="620" height="320" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="StartEvent_1_di" bpmnElement="StartEvent_1">
      <dc:Bounds x="182" y="162" width="36" height="36" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="Activity_19ctj9n_di" bpmnElement="Activity_0zxfjbm">
      <dc:Bounds x="270" y="140" width="100" height="80" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="Gateway_093kenq_di" bpmnElement="Gateway_093kenq"
isMarkerVisible="true">
      <dc:Bounds x="405" y="155" width="50" height="50" />
    <bpmndi:BPMNLabel>
      <dc:Bounds x="403" y="125" width="54" height="27" />
    </bpmndi:BPMNLabel>
    <bpmndi:BPMNShape>
    <bpmndi:BPMNShape id="Event_14ehgqs_di" bpmnElement="Event_14ehgqs">
      <dc:Bounds x="672" y="162" width="36" height="36" />
```

```
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_0qdt6yf_di" bpmnElement="Activity_16j2nbb">
  <dc:Bounds x="510" y="140" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Event_1e2p0uz_di" bpmnElement="Event_1e2p0uz">
  <dc:Bounds x="672" y="272" width="36" height="36" />
</bpmndi:BPMNShape>
<bpmndi:BPMNShape id="Activity_0qdtvew_di" bpmnElement="Activity_1mtzecx">
  <dc:Bounds x="510" y="250" width="100" height="80" />
</bpmndi:BPMNShape>
<bpmndi:BPMNEdge id="Flow_008de66_di" bpmnElement="Flow_008de66">
  <di:waypoint x="218" y="180" />
  <di:waypoint x="270" y="180" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_06boo6n_di" bpmnElement="Flow_06boo6n">
  <di:waypoint x="370" y="180" />
  <di:waypoint x="405" y="180" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1pl304m_di" bpmnElement="Flow_1pl304m">
  <di:waypoint x="610" y="180" />
  <di:waypoint x="672" y="180" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1xl8kpo_di" bpmnElement="Flow_1xl8kpo">
  <di:waypoint x="455" y="180" />
  <di:waypoint x="510" y="180" />
  <bpmndi:BPMNLabel>
    <dc:Bounds x="456" y="162" width="8" height="14" />
  </bpmndi:BPMNLabel>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_00j6ei0_di" bpmnElement="Flow_00j6ei0">
  <di:waypoint x="430" y="205" />
  <di:waypoint x="430" y="290" />
  <di:waypoint x="510" y="290" />
  <bpmndi:BPMNLabel>
    <dc:Bounds x="435" y="213" width="21" height="14" />
  </bpmndi:BPMNLabel>
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge id="Flow_1pdi1wf_di" bpmnElement="Flow_1pdi1wf">
  <di:waypoint x="610" y="290" />
  <di:waypoint x="672" y="290" />
</bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</bpmn:definitions>
```

Form

```
{
  "components": [
    {
      "label": "Nachname",
      "type": "textfield",
      "layout": {
        "row": "Row_1fp7fhr",
        "columns": null
      },
      "id": "Field_1t1zx8j",
      "key": "varLastname"
    },
    {
      "label": "Vorname",
      "type": "textfield",
      "layout": {
        "row": "Row_1alm5ht",
        "columns": null
      },
      "id": "Field_1esrj2e",
      "key": "varFirstname"
    }
  ],
  "type": "default",
  "id": "EinfachesFormular",
  "exporter": {
    "name": "Camunda Modeler",
    "version": "5.32.0"
  },
  "executionPlatform": "Camunda Platform",
  "executionPlatformVersion": "7.22.0",
  "schemaVersion": 18
}
```