

README zu den Downloadmaterialien

Ludger Schönfeld

Buch

Business Process Automation

Geschäftsprozesse nachhaltig automatisieren mit BPMN und DMN

von Ludger Schönfeld, Mitwirkung: Prof. Dr. Volker Stiehl

Rheinwerk Verlag, 2026

Stand

Dezember 2025

Kapitel 1

Einleitung

1.1 Download der Beispielmaterien

Die Beispielmaterien zum Buch finden Sie auf der Website des Rheinwerk-Verlags:

<https://www.rheinwerk-verlag.de/5716>

Sofern Sie das nicht bereits erledigt haben, laden Sie bitte das ZIP-Archiv auf Ihren Computer herunter und entpacken Sie es.

In dieser Anleitung gehe ich lediglich auf Artefakte aus den Materialien ein, die Sie zum Ausprobieren installieren müssen.

Hinweis 1: Demo – Prozessgesteuerte Anwendung »Seminarverwaltung«

Die hier von mir bereitgestellten Artefakte zur Prozessgesteuerten Anwendung »Seminarverwaltung« dienen lediglich der Veranschaulichung des Prozessgesteuerten Ansatzes. Die Anwendung ist nicht für den Produktivbetrieb geeignet. Es fehlen beispielsweise Sicherheitsaspekte.

Hinweis 2: Fokus der Anleitung: Betriebssystem »Windows«

Die Anleitung habe ich auf das Betriebssystem Windows ausgerichtet.

Allerdings nutze ich für die Software-Artefakte Container (Docker/Podman), sodass die Windows-spezifischen Schritte begrenzt sein sollten und Sie die Vorgehensweise für die anderen oben genannten Betriebssysteme relativ einfach übertragen können.

1.2 Übersicht über den Download-Ordner

Der Ordner mit den Begleitmaterialien ist nach Kapiteln strukturiert. Sie finden sich wie folgt zurecht:

Ordner	Erläuterung
Kapitel 2	Das Entscheidungsmodell »Verkaufspreis ermitteln« gibt es jeweils in den Varianten für <i>Camunda DMN Engine</i> und <i>Red Hat Drools/Kogito</i> .

Ordner	Erläuterung
Kapitel 3	Den Fachprozess »Seminaranmeldung abwickeln« habe ich mithilfe einer angedeuteten Verwendung des Camunda-BPMN-Frameworks modelliert.
Kapitel 5–6	<p>Zum Fachprozess »Seminaranmeldung abwickeln« finden Sie folgende Beispiel-Artefakte:</p> <ul style="list-style-type: none"> – Ergebnisdokumentation Prozesserhebungsworkshop – BPMN-Prozessmodelle – Entscheidungsmodell »Stornierungsgebühren ermitteln« – Skizze zur Benutzeraufgabe »Seminaranmeldungsanfrage erfassen« – Datenmodell
Kapitel 7	Enthält den BPMN-Integrationsprozess »Benachrichtigung versenden« und das Entscheidungsmodell »Korrespondenztemplate bestimmen« (SCIL).
Kapitel 8	<p>Die folgenden Softwareartefakte der PDA »Seminarverwaltung« finden Sie zum Ausprobieren:</p> <ul style="list-style-type: none"> – Artefakte zur Process-Engine <i>SpiffWorkflow</i>: Das sind die konfigurierten BPMN-Prozessmodelle für den implementierten Umfang sowie die API-Key-Erweiterung mit freundlicher Genehmigung des Herstellers der Process-Engine <i>Sartography</i>*, um die Process-Engine-Backend-API nutzen zu können. Bitte beachten Sie, dass diese Erweiterung lediglich in diesem Fallbeispiel verwendet und nicht weiterverbreitet werden darf. – Fachkomponente <i>PDA-SeminarAnmService</i> (»Dummy«) – <i>PDA-SemVerw-DecisionService</i> – PDA-SCIL-Integrationskomponente – Das Messaging-System <i>Apache ActiveMQ Classic</i>: Dieses ist nicht beigelegt, sondern das Image lässt sich per <i>docker-compose.yml</i>-Datei aus dem Docker Hub herunterladen. <p>Alle diese Komponenten lassen sich komfortabel mittels <i>docker-compose.yml</i> als Container aufsetzen.</p>

*Website: <https://sartography.com/>

Die physische Verteilungssicht der Artefakte zur Prozessgesteuerten Anwendung »Seminarverwaltung« finden Sie in Abbildung 1.1:

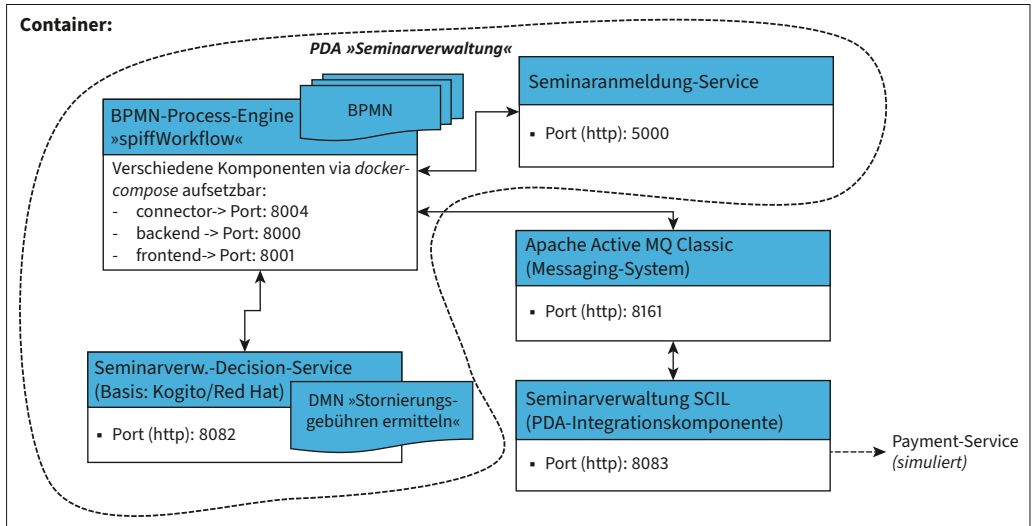


Abbildung 1.1: Übersicht über die physische Verteilungssicht der PDA »Seminarverwaltung« (Fallbeispiel)

Kapitel 2

Podman Desktop installieren und einrichten

Buchreferenz

- Kapitel 2
- Kapitel 8

Um Software innerhalb von Containern zu verwalten, werden Produkte wie *Podman* oder *Docker* verwendet. Durch die »Containerisierung« der Software wird das Aufsetzen des Fallbeispiels vereinfacht. Hier nutzen wir Podman.

Sie installieren und konfigurieren *Podman Desktop* wie folgt:

1. Öffnen Sie im Browser folgende URL: <https://podman-desktop.io/downloads>
2. Wählen Sie die für Ihr Betriebssystem passende Version aus, hier: Windows. Klicken Sie auf den zugehörigen Button **Download Now**. Sie laden damit die *.exe-Datei zu Podman Desktop herunter. Speichern Sie sich diese auf Ihrem Computer ab.
3. Klicken Sie auf die heruntergeladene *.exe-Datei zu Podman Desktop.
4. Folgen Sie der Installationsanleitung zu Podman unter <https://podman-desktop.io/docs/installation/windows-install>. An manchen Stellen finde ich die Anleitung von Podman zu ungenau. Deshalb habe ich Ihnen weiter unten ergänzende Schritte aufgeführt. Dies betrifft aus meiner Sicht *Schritt 1* und *Schritt 5* der Podman-Anleitung. Deshalb lesen Sie bitte parallel auch die nachfolgenden Schritte.

Hinweis 1: Voraussetzung – das CPU-Feature »Virtualisierung« muss unterstützt und aktiviert sein

Um *Podman* und Container zu nutzen, muss im BIOS das CPU-Feature der *Virtualisierung* aktiviert sein. Natürlich muss Ihre CPU das Feature der Virtualisierung auch unterstützen.

Wie Sie ins BIOS kommen, ist von Computer zu Computer unterschiedlich. Hierzu werden Sie im Internet fündig.




Im BIOS angelangt, finden Sie das Virtualisierungsfeature bei den Einstellungen der CPU. Die genaue Bezeichnung ist unterschiedlich. Manchmal heißt es einfach »Virtualization« oder »SVM Mode«. Ich empfehle Ihnen, im Handbuch zu Ihrer CPU nachzuschlagen oder einfach im Internet zu recherchieren.

Hinweis 2: Kommandozeilenbefehl eventuell mit Administrationsrechten ausführen

Je nach Computer und Ihren Benutzerberechtigungen müssen Sie die Kommandozeilenbefehle aus der Anleitung mit Administratorrechten ausführen. Öffnen Sie die Eingabeaufforderung unter Windows (Weiteres dazu in Schritt 5) mit Administrationsrechten. Klicken Sie dazu mit der rechten Maustaste auf die App **Eingabeaufforderung** und wählen Sie im Kontextmenü **Als Administrator ausführen** aus.

5. In der obigen Anleitung zu Podman müssen Sie WSL aktivieren. Dazu führen Sie die dort beschriebenen Kommandozeilenbefehle in der Eingabeaufforderung von Windows aus. Diese finden Sie über die Windows-Suche. Dort geben Sie »CMD« ein.

Tippen Sie Folgendes in der Eingabeaufforderung ein:

- `wsl --version` : Damit überprüfen Sie, ob bereits eine WSL-Version vorliegt. Wir brauchen hier die WSL-Version 2.
- `wsl --install --no-distribution` : Damit installieren Sie WSL, wenn dieses Feature noch nicht vorhanden ist.
- `wsl --set-default-version 2` : Damit setzen Sie die WSL-Version 2 als Standardversion. Diese Version brauchen wir unbedingt für Podman.

6. Starten Sie danach Ihren Computer neu.

7. Öffnen Sie *Podman Desktop* und klicken Sie auf den Button **Set up** direkt im Dashboard (siehe [Abbildung 2.1](#)).

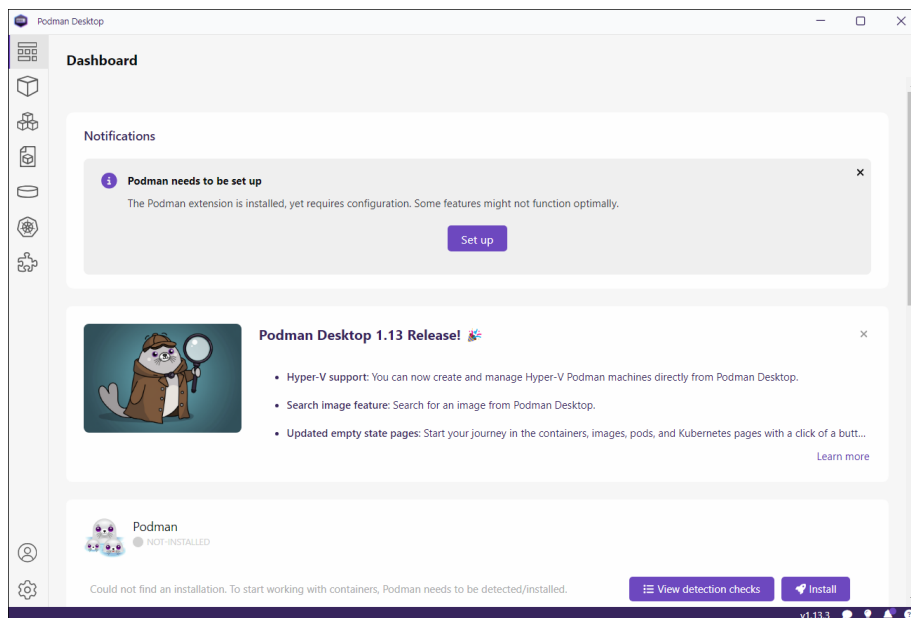


Abbildung 2.1: Podman Desktop – Startbildschirm mit der »Set up«-Option

8. Im nun erscheinenden Setup-Dialog stellt die Anwendung fest, dass kein Podman gefunden werden konnte. Deshalb klicken Sie auf den Button **Next** (siehe Abbildung 2.2).

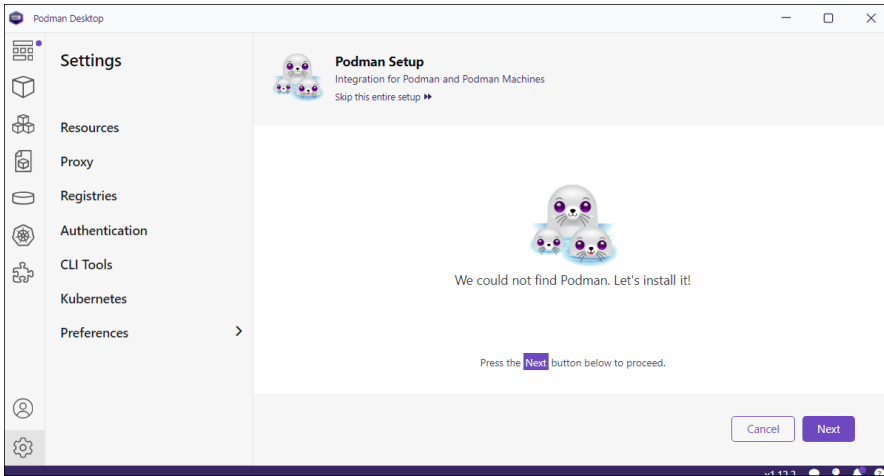


Abbildung 2.2: Podman Desktop – das Podman-Setup stellt fest, dass kein Podman gefunden werden konnte.

9. Im nächsten Schritt ermittelt das Podman-Setup, welche Anforderungen noch erfüllt werden müssen. In Abbildung 2.3 sehen Sie ein Beispiel dafür. Erläuterungen zum weiteren Vorgehen finden Sie im vorherigen Schritt zu WSL und in den nachfolgenden Schritten dieser Anleitung.

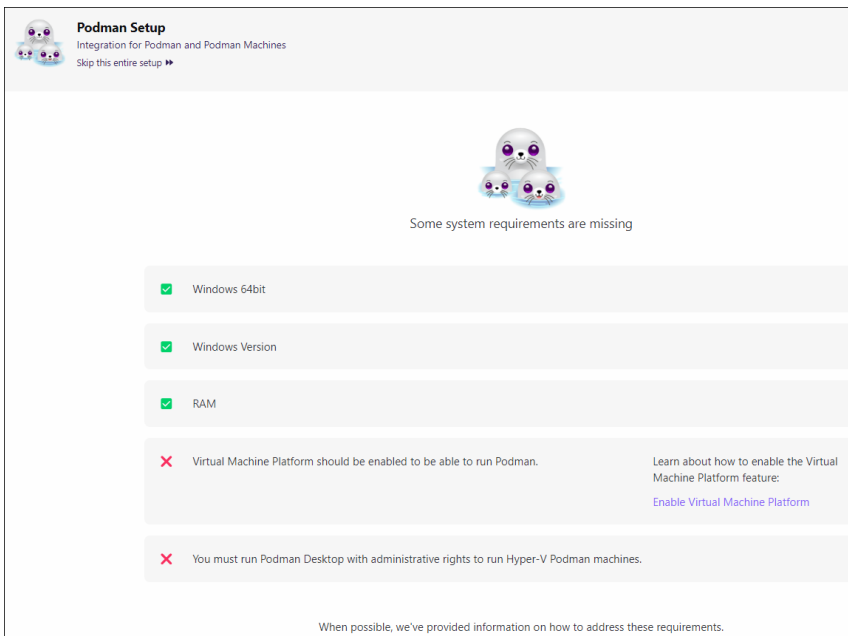


Abbildung 2.3: Podman Desktop – Podman Setup – Auflistung fehlender Anforderungen

10. Aktivieren Sie das Hyper-V-Feature. Wie Sie Hyper-V aktivieren, erfahren Sie auf folgender Microsoft-Seite:

<https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>

Sie können Hyper-V entweder über die Eingabeaufforderung oder über die PowerShell aktivieren. Anschließend müssen Sie den Computer neu starten.

11. Öffnen Sie *Podman Desktop* erneut und vergewissern Sie sich, dass die Installation erfolgreich war.
12. Nun müssen Sie eine *Podman machine* anlegen. Klicken Sie im linken Bereich weiter unten auf das Icon **Settings**. In dem Tab **Resources** klicken Sie im Bereich *Podman* auf den Button **Create new**. Daraufhin wird Ihnen der Dialog **Create a Podman machine** angezeigt (siehe Abbildung 2.4).

Create a Podman machine

Name:
podman-machine-default

Image Path (Optional):
[Folder icon]

Image URL or image reference (Optional):
Use 'registry/org/image:version' for image reference

Machine with root privileges:
Enabled [Toggle]

User mode networking (traffic relayed by a user process). See [documentation](#).
Disabled [Toggle]

Start the machine now:
Enabled [Toggle]

Create

Abbildung 2.4: Podman Desktop – Erzeugen einer neuen »Podman machine«

13. Im Dialog **Create a Podman machine** können Sie die in Abbildung 2.4 dargestellten Standardeinstellungen so ohne Änderung beibehalten. Abschließend klicken Sie auf den Button **Create**.

Hinweis 3: Der Fehler »Error: Command execution failed with exit code 125 Command execution failed with exit code 125 Error: hyperv machines require admin authority«

Manchmal bekommen Sie die oben genannte Fehlermeldung. Zur Behebung müssen Sie Podman Desktop mit Administrationsrechten ausführen.

Zum Starten von Podman Desktop klicken Sie mit der rechten Maustaste auf das Icon von Podman Desktop und wählen im Kontextmenü **Als Administrator ausführen** aus.

14. Nach erfolgreicher Installation und Konfiguration von Podman Desktop brauchen wir noch die *Compose Extension*. Dazu klicken Sie in Podman Desktop innerhalb des linken Bereichs mit den Icons auf das Icon zu **Extensions**.

Daraufhin erscheint die Liste mit den Extensions. Suchen Sie die **Compose extension**. Klicken Sie bei der **Compose extension** auf den Button mit dem **Zahnrad** und folgen Sie den Anweisungen zur Einrichtung. Der Assistent stellt fest, dass diese Extension noch nicht vorhanden ist und installiert werden muss. Klicken Sie dazu auf den Button **Next**. Nach einem kurzen Augenblick ist die Installation abgeschlossen.

Damit haben Sie Podman Desktop für den hier erforderlichen Zweck erfolgreich eingerichtet.

Sollte Podman im Dashboard *nicht* als **Running** angezeigt werden, müssen Sie auf den **Start**-Button klicken, um Podman zu starten.

Hinweis 4: Einfaches Troubleshooting


Problem

Sie öffnen Podman Desktop und sehen, dass die vorhandenen Container und Images nicht angezeigt werden. Die jeweiligen Ansichten sind leer.

Manchmal dauert es ein bisschen, bis die vorhandenen Images oder Container in der jeweiligen Ansicht angezeigt werden. Was können Sie dann tun?

- ein paar Minuten Geduld haben
- Podman Desktop vollständig beenden. Das bedeutet auch, in der Taskleiste unter den ausgeblendeten Symbolen Podman Desktop zu beenden.
- Dann starten Sie Podman Desktop erneut. Manchmal hilft es, auf der Eingabeaufforderung die folgenden Befehle einzugeben:

```
podman machine stop 
```

```
podman machine start 
```


Kapitel 3

Das Entscheidungsmodell »Verkaufspreis ermitteln« ausführen

Buchreferenz

Kapitel 2

Sie finden im Verzeichnis zum Entscheidungsmodell *Verkaufspreis ermitteln* Varianten für *Drools* von RedHat und für die *Camunda DMN Engine* vor.

Ich zeige Ihnen, wie Sie relativ schnell innerhalb einer Demo-Umgebung die verschiedenen Varianten ausprobieren können.

3.1 Demo-Umgebung für Camunda

Am einfachsten nutzen Sie hierfür die *Camunda 8 SaaS-Umgebung* mit *Starter Trial*-Lizenz.

1. Erstellen Sie einen Account unter folgender URL:

<https://signup.camunda.com/accounts/>

2. Nach erfolgreicher Registrierung loggen Sie sich in Ihre Umgebung unter folgender URL ein: <https://weblogin.cloud.camunda.io/>. Nach erfolgreichem Login befinden Sie sich in der Web-App *Console*. Diese App können Sie im oberen rechten Bereich erkennen. Wenn Sie sich nicht dort wiederfinden, klicken Sie auf die Punkte im oberen linken Bereich und wählen im danach erscheinenden Menü die App *Console* aus.

3. Navigieren Sie zum Bereich **Clusters** innerhalb der *Console*.

4. Legen Sie nun einen Cluster an. Dazu folgen Sie bitte den Erklärungen zum Anlegen eines Clusters. Nach einem kurzen Moment Geduld wird der Cluster gestartet.

5. Wechseln Sie nun über die oberen Punkte zur Web App *Modeler*.

6. Legen Sie über den Button **New Project** oder den Link **Create new project** ein neues Projekt an. Den Projektnamen tragen Sie im oberen Navigationspfad des geöffneten Projektordners ein: »Verkaufspreis ermitteln«.

7. Danach klicken Sie auf dem Pfeil nach unten bei **Create new** und wählen im Menü den Eintrag **Upload files** aus (siehe [Abbildung 3.1](#)).

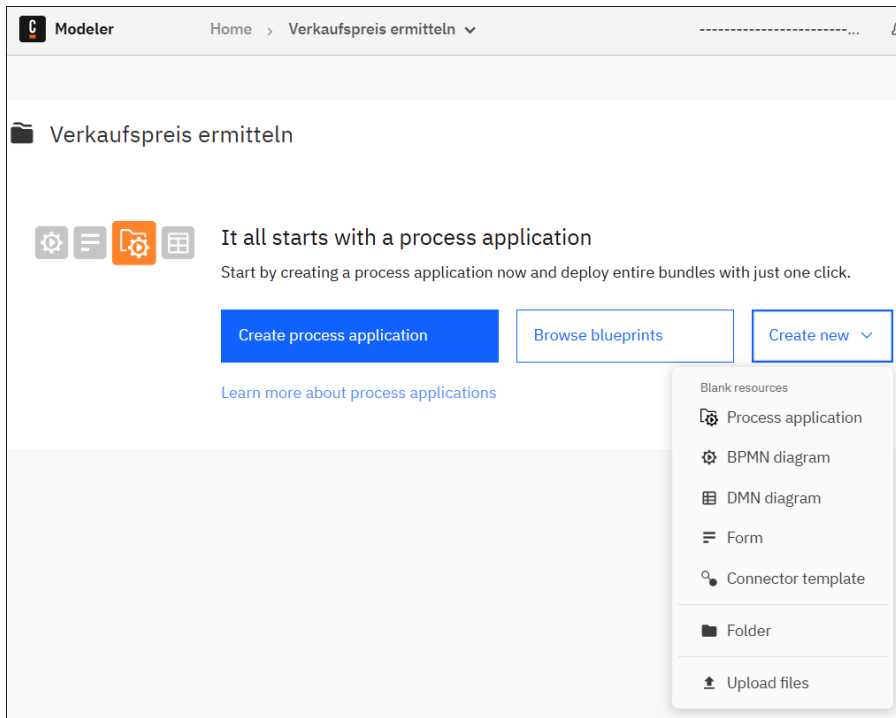


Abbildung 3.1: Leeres Projekt »Verkaufspreis ermitteln« in »Camunda 8 SaaS« (Web-App »Modeler«)

8. Wählen Sie die Dateien zum Entscheidungsmodell »Verkaufspreis bestimmen« für Camunda 8 aus den Downloadmaterialien aus. Sie finden diese unter dem Verzeichnis zum oben genannten Kapitel.
9. Öffnen Sie die DMN-Datei, indem Sie diese anklicken. Sie gelangen nun in den Editor. Klicken Sie im oberen rechten Bereich auf den Button **Deploy**. Damit wird das Entscheidungsmodell im Cluster deployt.
Gehen Sie genauso auch für den Testprozess vor. Öffnen Sie diesen und klicken Sie auf den oberen rechten Button **Deploy**. Mit dem Testprozess wird auch automatisch das verknüpfte Formular für die User Task deployt.
10. Wechseln Sie nun über die oberen Punkte zur Web-App *Tasklist*. Wechseln Sie dort wiederum zum Bereich **Prozesse**, indem Sie auf den gleichnamigen Tab klicken. Wir wollen das Entscheidungsmodell im Rahmen des Testprozesses ausprobieren.
11. Klicken Sie auf den Prozess **Testproc_Verkaufspreis ermitteln**. Damit starten Sie eine Prozessinstanz zu diesem Prozess (siehe Abbildung 3.2).

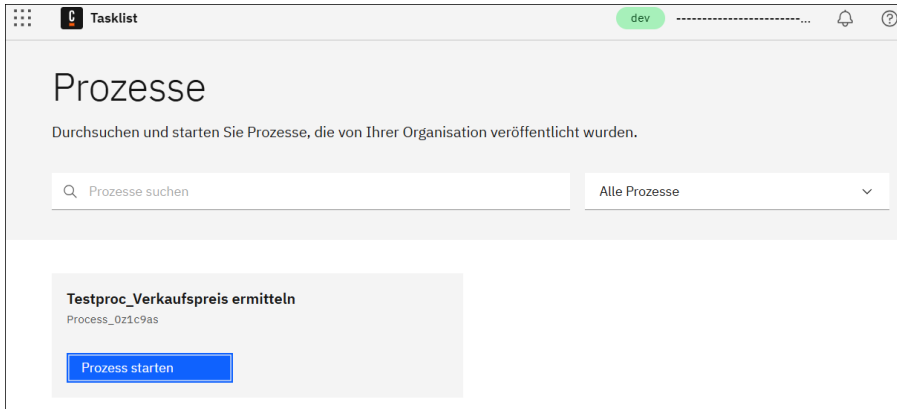


Abbildung 3.2: Eine Prozessinstanz aus Anwendersicht zum Testprozess starten

12. Unter dem oberen Tab **Aufgaben** wird Ihnen eine Aufgabe angezeigt. Wechseln Sie nun bitte zu diesem Tab.
13. Klicken Sie auf die erscheinende Aufgabe. Sie sehen im rechten Bereich das Formular zu dieser User Task.
14. Klicken Sie auf den oberen Button **Mir zuweisen**. Danach können Sie das Formular bearbeiten. Geben Sie die in Abbildung 3.3 dargestellten Werte ein. Anschließend klicken Sie auf den Button **Aufgabe abschliessen**.

Abbildung 3.3: User Task »Produktangaben definieren« (Beispiel: Produktkategorie »Dienstleistung«)

15. Danach erscheint eine weitere Aufgabe in der Taskliste: **Ergebnis Verkaufskalkulation anzeigen**. In dieser User Task finden Sie alle in der Prozessinstanz vorhandenen Prozessvariablen. Neben den vorher eingegebenen Daten finden Sie auch

die Prozessvariable *grossSalesPrice*. Diese zeigt Ihnen das Ergebnis der Berechnung aus der Evaluation des Entscheidungsmodells *Verkaufspreis ermitteln* (siehe Abbildung 3.4).

Ergebnis Verkaufspreis-kalkulation anzeigen

Testproc_Verkaufspreis ermitteln

Nicht zugewiesen

Mir zuweisen

Aufgabe

Prozess

Variablen

Variable hinzufügen +

Name	Wert
grossSalesPrice	1190
listSalesPrice	1000
product	"DMN Training"
productGroup	"SERVICE"

Aufgabe abschließen

Abbildung 3.4: Ergebnis der Verkaufspreiskalkulation anzeigen (Beispiel)

16. Um den Testprozess abzuschließen, klicken Sie auf den Button **Mir zuweisen**. Anschließend können Sie diese User Task abschließen. Dazu klicken Sie auf den Button **Aufgabe abschliessen**.

Damit haben Sie die Evaluation des Entscheidungsmodells in der *Camunda 8*-Umgebung einmal praktisch erlebt. Wenn Sie wollen, können Sie gleich nochmals eine Prozessinstanz des Testprozesses starten. Dann können Sie beispielsweise einmal die Berechnung anhand der Produktkategorie »Buch« erleben.

3.2 Demo-Umgebung für Drools (Red Hat)

3.2.1 Voraussetzung

Sie haben *Podman Desktop* installiert und konfiguriert (vgl. Abschnitt 2, »Podman Desktop installieren und einrichten«).

3.2.2 Container zum Image »jBPM Full server« erzeugen und für Demo einrichten

1. Öffnen Sie die Anwendung *Podman Desktop*.
2. Klicken Sie am linken Rand auf das Symbol für **Images**.

3. Nun klicken Sie auf den oberen Button **Pull** und geben anschließend folgenden Image-namen ein: `quay.io/kiegroup/jbpm-server-full`¹ (siehe [Abbildung 3.5](#)).

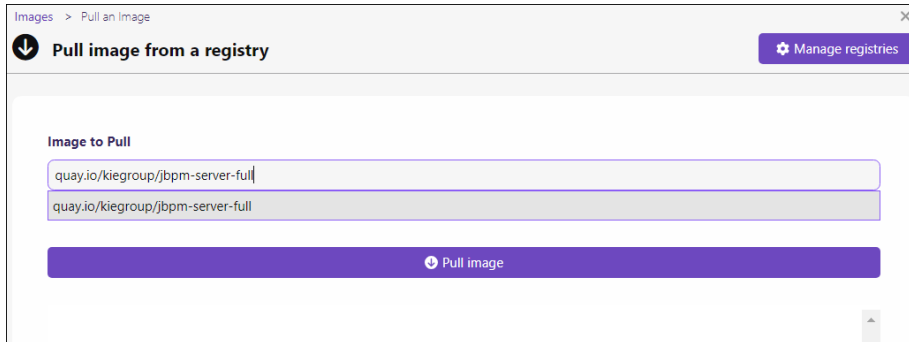


Abbildung 3.5: Podman Desktop – das Image »jbpm-server-full« herunterladen

4. Zum Herunterladen des Images klicken Sie auf den Button **Pull image**. Nun wird das Herunterladen des Images gestartet. Den Vorgang können Sie nun sehen. Nach Abschluss werden Ihnen die Meldungen **Download complete** angezeigt und der Button **Done** erscheint (siehe [Abbildung 3.6](#)).

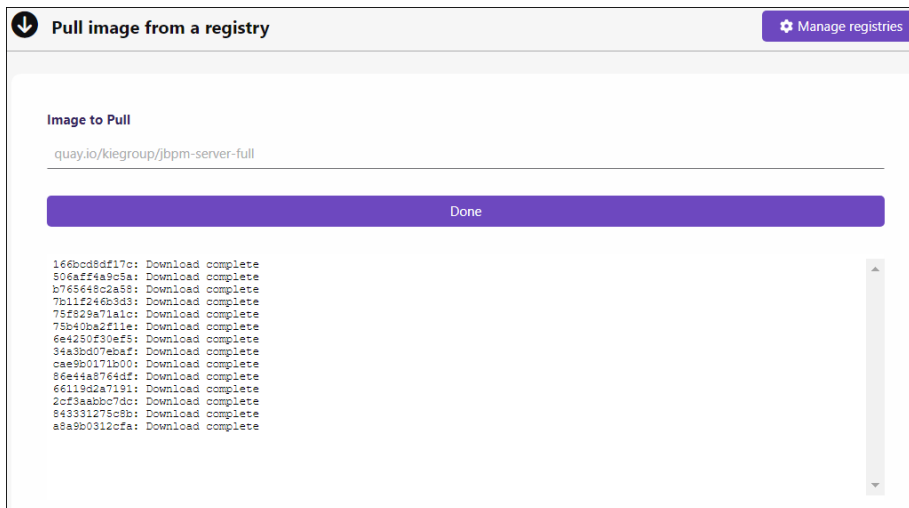


Abbildung 3.6: Podman Desktop – Download des Image »jbpm-server-full« abgeschlossen

5. Klicken Sie auf den Button **Done**. Sie gelangen zum Bereich **Images** zurück. Dort finden Sie nun auch das Image `jbpm-server-full`.

¹ Genaue Beschreibung zum Images finden Sie unter: <https://quay.io/repository/kiegroup/jbpm-server-full>

6. Sie erzeugen und starten einen Container, indem Sie auf das **Play**-Symbol beim Image klicken. Nun erscheint ein Dialog, um den Container zu konfigurieren. Behalten Sie die Standardeinstellungen bei und klicken Sie auf den Button **Start Container** (siehe Abbildung 3.7).

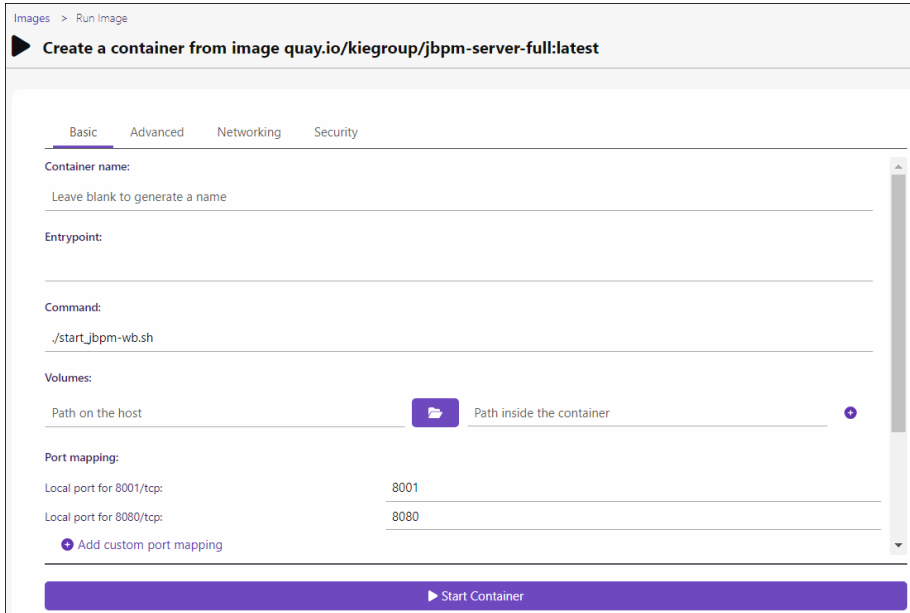


Abbildung 3.7: Podman Desktop – Container zum Image »jbpm-server-full« konfigurieren und starten

7. Der Container wird mit der Anwendung hochgefahren. Sie können die Logs sehen.
8. Öffnen Sie Ihren Browser und tippen Sie folgende URL ein: <http://localhost:8080/business-central>.
9. Loggen Sie sich wie folgt in die Benutzeroberfläche *Business Central* ein:
 - Benutzer: wbadmín
 - Passwort: wbadmín
10. Auf der Willkommensseite von *Business Central* klicken Sie unter dem Bereich **Design** auf den Link **Projekte**.
11. Anschließend müssen Sie einen *Space* einrichten. Wir nutzen hier den Standard-Space **MySpace**. Klicken Sie bitte auf diesem Space.

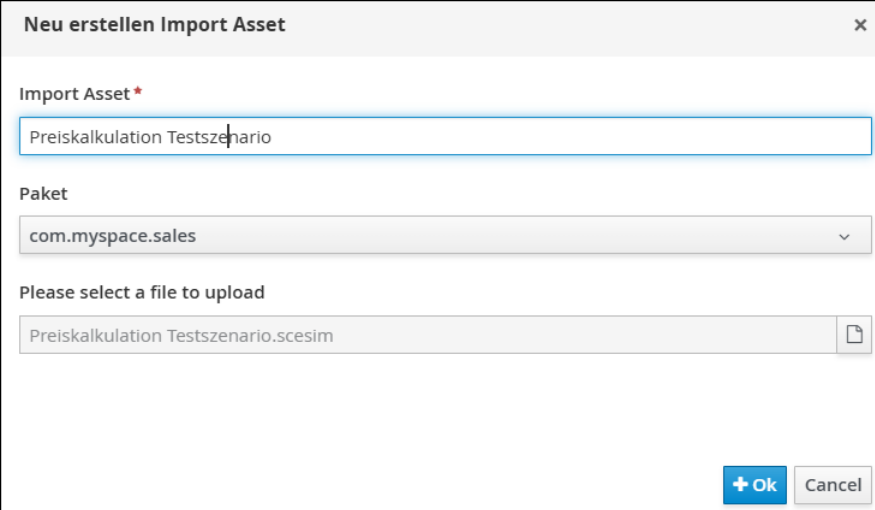
12. Klicken Sie innerhalb von **MySpace** auf den Button **Add Project**. Geben Sie nun als Projektnamen »Sales« an. Abschließend klicken Sie auf **Hinzufügen**:
13. Nun befinden Sie sich im angelegten Projekt *Sales*. Über den Button **Import Asset** importieren Sie nun die nachfolgend genannten Dateien aus dem Download-Materialien zur Variante *Drools-RedHat* des Entscheidungsmodells *Verkaufspreis ermitteln*. Sie können leider immer nur eine Datei gleichzeitig ins Projekt hochladen. Deshalb müssen Sie diesen Vorgang für alle unten genannten Dateien wiederholen:
 - *Verkaufspreis bestimmen.dmn*
 - *Preiskalkulation Testszenario.scesim*

Anmerkung: Später können Sie auch die Dateien aus dem Unterordner *Boxed Context-Variante Bruttopreis kalkulieren\jBPM-Drools* importieren. Hier befindet sich eine Variante des Entscheidungsmodells. Dazu müssen Sie dieselben Schritte zum Hochladen in das Projektverzeichnis durchführen.

Im Dialog **Neu erstellen Import Asset** geben Sie die Angaben jeweils für die oben genannten Dateien an (siehe [Abbildung 3.8](#) und [Abbildung 3.9](#)) und klicken anschließend auf den Button **+Ok**. Wenn Sie jeweils eine Datei hochgeladen haben, wird Ihnen der Inhalt direkt angezeigt. Damit Sie eine weitere Datei hochladen können, müssen Sie zum Projektordner zurückkehren. Dazu klicken Sie einfach im oberen Navigationspfad auf **Sales**.

The screenshot shows a dialog box titled "Neu erstellen Import Asset". It has a close button (X) in the top right corner. The dialog is divided into three main sections. The first section, "Import Asset", has a red asterisk and a text input field containing "Verkaufspreis bestimmen". The second section, "Paket", has a dropdown menu showing "com.myspace.sales". The third section, "Please select a file to upload", has a text input field containing "Verkaufspreis bestimmen.dmn" and a file icon button. At the bottom right, there are two buttons: "+ Ok" and "Cancel".

Abbildung 3.8: jBPM Business Central – das DMN-Entscheidungsmodell »Verkaufspreis bestimmen« hochladen



The screenshot shows a dialog box titled "Neu erstellen Import Asset". It has a close button (X) in the top right corner. The dialog contains three sections: "Import Asset *" with a text input field containing "Preiskalkulation Testszenario"; "Paket" with a dropdown menu showing "com.myspace.sales"; and "Please select a file to upload" with a file input field showing "Preiskalkulation Testszenario.scesim" and a file icon. At the bottom right are "+ Ok" and "Cancel" buttons.

Abbildung 3.9: jBPM Business Central – das Testszenario »Preiskalkulation Testszenario.scesim« hochladen

Damit haben Sie nun die Demo-Umgebung erfolgreich aufgesetzt. Im folgenden Abschnitt zur Evaluation des Entscheidungsmodells erläutere ich, wie Sie das Entscheidungsmodell mithilfe des Testszenarios evaluieren.

3.2.3 Evaluation des Entscheidungsmodells

Sie evaluieren das Entscheidungsmodell mithilfe des Testszenario wie folgt:

1. Öffnen Sie das weiter oben importierte Testszenario *Preiskalkulation Testszenario*.
2. Klicken Sie im oberen Bereich auf den **Play**-Button. Damit wird das Testszenario vollständig ausgeführt.
3. Im erscheinenden **Test Report** sehen Sie, dass das definierte Testszenario erfolgreich war (siehe [Abbildung 3.10](#)).

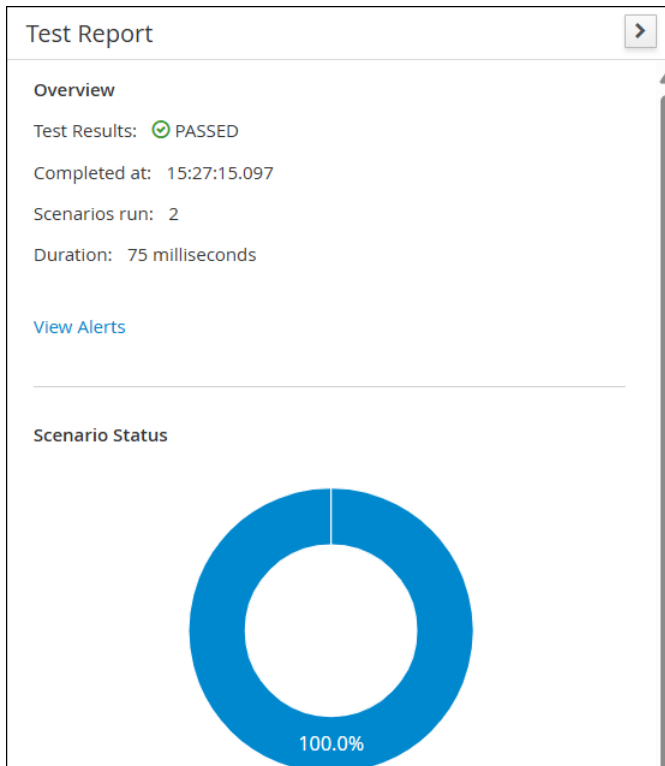


Abbildung 3.10: »Test Report« – erfolgreiche Testausführung

4. Wenn Sie sehen möchten, wie der Test Report im Fall von Fehlern aussieht, gehen Sie wie folgt vor: Geben Sie in dem Bereich **Expect** der Testszenario-Tabelle beispielsweise beim Szenario »Bücher« in der Spalte **Bruttopreis kalkulieren** den Wert 45 ein.
5. Führen Sie mithilfe des **Play**-Buttons das Testszenario erneut aus. Nun sehen Sie, dass der Fall »Bücher« fehlerhaft war – auch anhand der farbigen Hervorhebung der Zelle. Wenn Sie mit der Maus über die Zelle gehen, wird Ihnen auch der ermittelte Wert angezeigt (siehe Abbildung 3.11).

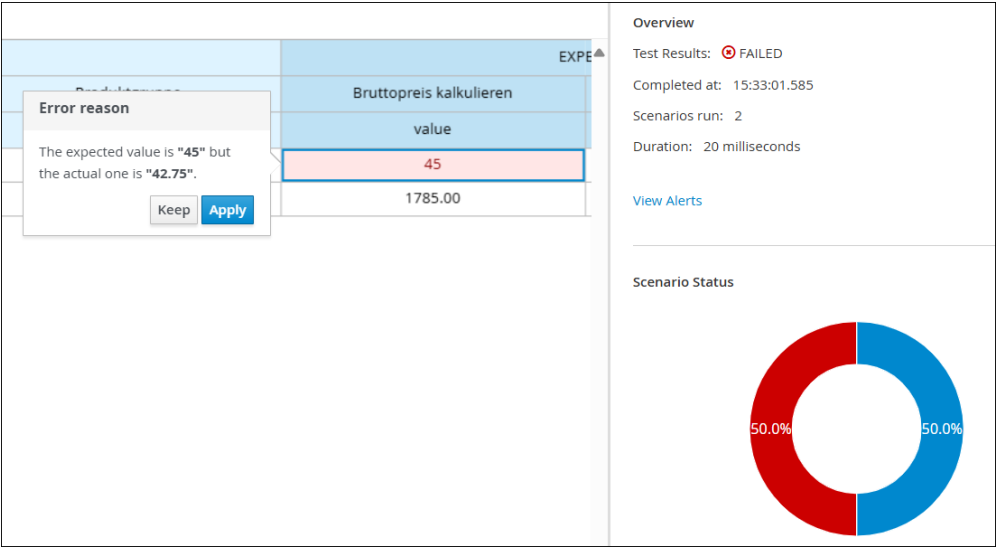


Abbildung 3.11: Test Report – fehlerhaftes Testergebnis

Nun haben Sie das Entscheidungsmodell *Verkaufspreis bestimmen* mithilfe des Produkts *Drools* von Red Hat ausprobiert. Sie können noch ein wenig weiter ausprobieren oder die Umgebung herunterfahren. Mehr dazu lesen Sie im folgenden Abschnitt.

3.2.4 Den Container »JBPM Full server« herunterfahren

- 1. Loggen Sie sich aus der Oberfläche *Business Central* aus.
- 2. Gehen Sie in *Podman Desktop* auf den Bereich **Containers**. Klicken Sie auf das **Stop Container**-Symbol beim Container *jbpm-server-full* (siehe Abbildung 3.12).

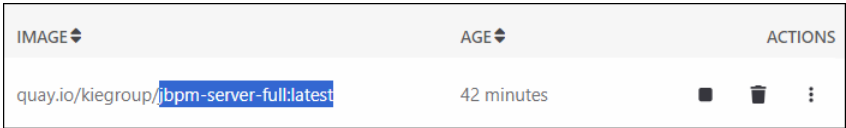


Abbildung 3.12: Podman Desktop – Bereich »Containers« – den Container »jbpm-server-full« herunterfahren

Der Container wird heruntergefahren.

Wenn Sie den Container nicht mehr benötigen, dann können Sie ihn auch über das Symbol **Papierkorb** löschen.

Kapitel 4

Fallstudie:

PDA »Seminarverwaltung« ausführen

Buchreferenz

Kapitel 8

4.1 Voraussetzung

Sie haben *Podman Desktop* installiert und konfiguriert (vgl. Kapitel 2, »Podman Desktop installieren und einrichten«).

4.2 Externes Netzwerk für Container in Podman erstellen

Damit die verschiedenen Container miteinander kommunizieren können, müssen sich alle Container in demselben Netzwerk befinden. Dazu müssen Sie ein externes Netzwerk für Ihre Podman-Umgebung einrichten.

So gehen Sie vor:

1. In der Windows-Suchleiste am unteren Bildschirmrand tippen Sie »CMD« ein. Klicken Sie die App *Eingabeaufforderung* an, die in den Suchergebnissen erscheint.
2. In der Eingabeaufforderung geben Sie folgenden Befehl ein:

```
podman network create pda-seminarverwaltung
```

4.3 Die BPMN-Process-Engine »SpiffWorkflow« aufsetzen

1. Legen Sie einen neuen Ordner *spiffworkflow* nahe an der Wurzel des Laufwerks oder der Partition Ihrer Festplatte an. Zum Beispiel so: *D:\spiffworkflow*.
2. Öffnen Sie die folgende URL im Browser: <https://github.com/sartography/spiff-arena/blob/main/docker-compose.yml>. Hier finden Sie die *docker-compose*-Datei von Spiff-Workflow.

Hinweis: Beigefügter SpiffWorkflow »docker-compose.yml«

Ich habe bereits eine *docker-compose.yml*-Datei für SpiffWorkflow mit allen unten beschriebenen Änderungen den Download-Materialien hinzugefügt. Sie brauchen also nicht unbedingt diese Datei erneut herunterzuladen. Wenn Sie die beigefügte Datei direkt nutzen, brauchen Sie keine der beschriebenen Änderungen an dieser *docker-compose.yml* vorzunehmen und können die Schritte unter 3. überspringen.

Sollten Sie die aktuelle Datei vom Hersteller von SpiffWorkflow nutzen müssen, finden Sie alle Änderungen weiter unten beschrieben.

Laden Sie diese Datei in das unter Schritt 1 angelegte Verzeichnis herunter, indem Sie rechts bei den kleinen Buttons auf den Button **Download raw file** klicken und dann den entsprechenden Ordner auf Ihrer Festplatte auswählen.

3. Öffnen Sie die *docker-compose.yml* von SpiffWorkflow und ergänzen Sie sie wie folgt:
 - Unter den Umgebungsvariablen (*environment*) bei der Backend-Komponente ergänzen Sie folgende Zeile (siehe [Abbildung 4.1](#)):

```
SPIFFWORKFLOW_BACKEND_EXTENSIONS_API_ENABLED: "true"
```

Damit aktivieren Sie die Backend-API-Erweiterung. Die Dateien der Erweiterung werden Sie im nächsten Schritt installieren.

```
spiffworkflow-backend:
  container_name: spiffworkflow-backend
  image: ghcr.io/sartography/spiffworkflow-backend:latest
  # Enable setting the ownership of created files to match
  # RUN_AS="$(id -u):$(id -g)" docker compose up -d
  user: ${RUN_AS:-0:0}
  environment:
    SPIFFWORKFLOW_BACKEND_EXTENSIONS_API_ENABLED: "true"
    SPIFFWORKFLOW_BACKEND_ENV: "local_docker"
    FLASK_DEBUG: "0"
```

Abbildung 4.1: BPMN-Process-Engine »SpiffWorkflow« – Umgebungsvariable für die Aktivierung der Backend-API-Extension in der »docker-compose«-Datei ergänzen

- Am Ende der vorhandenen Konfigurationszeilen zu *jeder* der Komponenten ergänzen Sie jeweils die Netzwerkverwendung. Exemplarisch habe ich Ihnen dies für die Komponente *spiffworkflow-connector* in [Abbildung 4.2](#) gezeigt.


```

spiffworkflow-connector:
  container_name: spiffworkflow-connector
  image: ghcr.io/sartography/connector-proxy-demo:latest
  environment:
    FLASK_ENV: "${FLASK_ENV:-development}"
    FLASK_DEBUG: "0"
    FLASK_SESSION_SECRET_KEY: "${FLASK_SESSION_SECRET_KEY:-super_secret_key}"
    CONNECTOR_PROXY_PORT: "${SPIFF_CONNECTOR_PORT:-8004}"
  ports:
    - "${SPIFF_CONNECTOR_PORT:-8004}:${SPIFF_CONNECTOR_PORT:-8004}/tcp"
  healthcheck:
    test: "curl localhost:${SPIFF_CONNECTOR_PORT:-8004}/liveness --fail"
    interval: 10s
    timeout: 5s
    retries: 20
  networks:
    - pda-seminarverwaltung

```

Abbildung 4.2: BPMN-Process-Engine »SpiffWorkflow« – Ergänzung der Angabe der Netzwerkverwendung in der »docker-compose.yml« zur Komponente »spiffworkflow-container« (Beispiel)

```
networks:
```

```
  - pda-seminarverwaltung
```

- Bitte beachten Sie hier die Einrückungen. `networks` muss genau auf derselben Höhe wie beispielsweise `ports` sein. Die Angabe `- pda-seminarverwaltung` bekommt zwei Einrückungen in Form von Leerzeichen, ausgehend von `networks`.
- Am Ende der Datei ergänzen Sie die Referenzierung des externen Netzwerks. Achten Sie dabei bitte unbedingt auf die Einrückungen: `networks` erhält keine Einrückung von `links`; `pda-seminarverwaltung` wird ausgehend von `networks` zwei Leerzeichen nach rechts verschoben, und `external` wird ausgehend von `pda-seminarverwaltung` zwei Leerzeichen nach rechts verschoben:

```
networks:
```

```
  pda-seminarverwaltung:
```

```
    external: true
```

- Abschließend speichern Sie die Änderungen in der Datei ab: z. B. mit `Strg` + `S` und schließen die Datei.
4. Legen Sie ein Unterverzeichnis im Ordner *spiffworkflow* an. Dieses Verzeichnis muss *process_models* heißen. Hierhin werden später die Prozessmodelle aus einem der SpiffWorkflow-Container ausgelagert.
 5. Nehmen Sie den gesamten Ordnerinhalt unter *Kapitel 8/00_Process Engine SpiffWorkflow/process_models* aus den Downloadmaterialien zum Buch und verschieben Sie diesen in das Verzeichnis *process_models* auf Ihrem Computer. Hier finden Sie die konfigurierten BPMN-Prozesse zum implementierten Umfang der Fallstudie und die API-Key-Erweiterung von SpiffWorkflow.

6. Öffnen Sie nun die Eingabeaufforderung, indem Sie im Suchfeld in der Taskleiste »CMD« eintippen. Klicken Sie die App *Eingabeaufforderung* in der Trefferliste an. Öffnen Sie die App bitte mit Administratorrechten.
7. In der Eingabeaufforderung führen Sie nacheinander folgende Befehle aus:
- `cd D:/spiffWorkflow` [↩]¹ – Sie müssen sich mit der Eingabeaufforderung innerhalb des *spiffWorkflow*-Verzeichnisses befinden.
 - `docker-compose pull` [↩] – Die Images für SpiffWorkflow werden aus dem Internet heruntergeladen.
 - `docker-compose up -d` [↩] – Damit starten Sie SpiffWorkflow mit all seinen Komponenten.

Das Ergebnis der gestarteten Container der SpiffWorkflow-Komponenten können Sie in *Podman Desktop* im Bereich **Containers** kontrollieren (siehe Abbildung 4.3).

Sie sehen dort eine Angabe des Alters von den laufenden Containern sowie die Buttons **Stop Container** (die kleinen Quadrate links neben dem Mülleimer-Symbol).

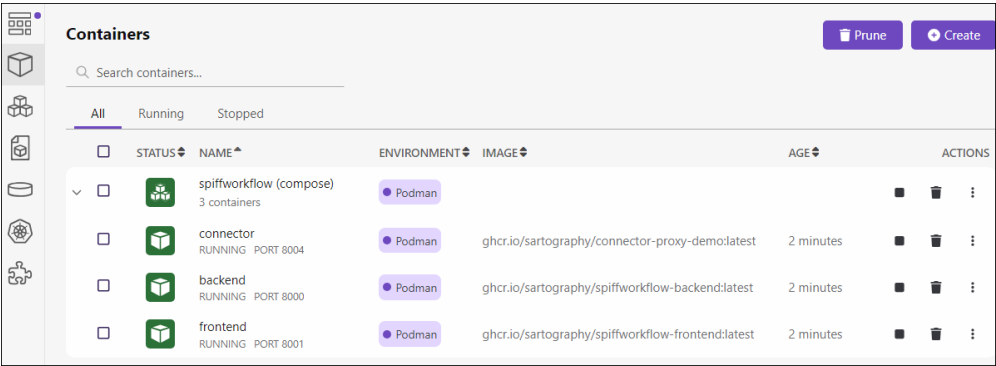


Abbildung 4.3: BPMN-Process-Engine »SpiffWorkflow« – laufende Container der BPMN-Process-Engine »SpiffWorkflow« in Podman

8. Auf das SpiffWorkflow-Frontend gelangen Sie, indem Sie im Browser die folgende URL eingeben: `http://localhost:8001`. Sie melden sich an, indem Sie als Benutzer »admin« und als Passwort »admin« eintippen und anschließend auf die Schaltfläche **Log In** klicken.

1 Sollte Ihre Standardpartition bzw. Ihr Standardlaufwerk ein anderes sein, beispielsweise C:/, müssen Sie zunächst zur Partition bzw. zum Laufwerk D:/wechseln. Tippen Sie dazu in der Eingabeaufforderung ein: `D: [↩]`

9. Nach dem Einloggen klicken Sie rechts oben auf Ihr Profil **Admin**. Dann öffnet sich ein Menü. Dort klicken Sie auf den Eintrag **API keys** (siehe [Abbildung 4.4](#)). Der Menüeintrag **API keys** wird nur angezeigt, wenn die API-Key-Extension aktiv ist. Sie wurde bereits in Schritt 3 über die docker-compose-Konfiguration des Process-Engine-Backends aktiviert und über den Ordner *process_models* in Schritt 5 installiert.

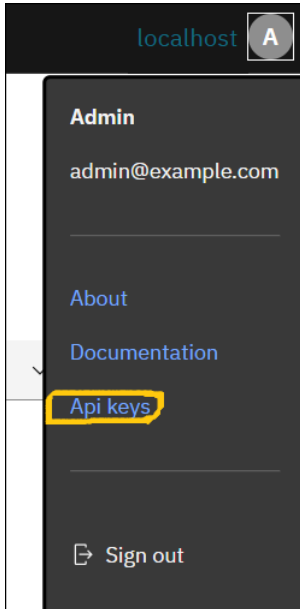


Abbildung 4.4: BPMN-Process-Engine »SpiffWorkflow« – Frontend: die Backend-API-Keys

10. Im nächsten Schritt geben Sie einen beschreibenden Namen für den zu erzeugenden Backend-API-Key ein. Zum Beispiel: »pda-scil«. Klicken Sie danach auf den Button **Submit**.

Nun wird Ihnen der generierte API-Key angezeigt. Dieser Key wird Ihnen nur einmal angezeigt. Deshalb müssen Sie sich diesen Key merken (oder ihn abspeichern). Sie brauchen ihn später.

11. Prüfen Sie, ob die relevanten Fachprozesse für die Fallstudie vorhanden sind. Dazu klicken Sie im oberen Navigationsmenü auf **Processes**. Sie sollten nun eine Prozessgruppe **case-study-seminarverwaltung** sehen. Wenn Sie auf diese Prozessgruppe klicken, finden Sie drei Prozessmodelle: *Globaler Prozess: Zahlungsbeleg erstellen*, *Seminaranmeldung abwickeln* und *Seminaranmeldungsanfrage abgeben* (siehe [Abbildung 4.5](#)). Wenn Sie jeweils auf die Prozessmodelle klicken, finden Sie die zugehörigen Dateien.

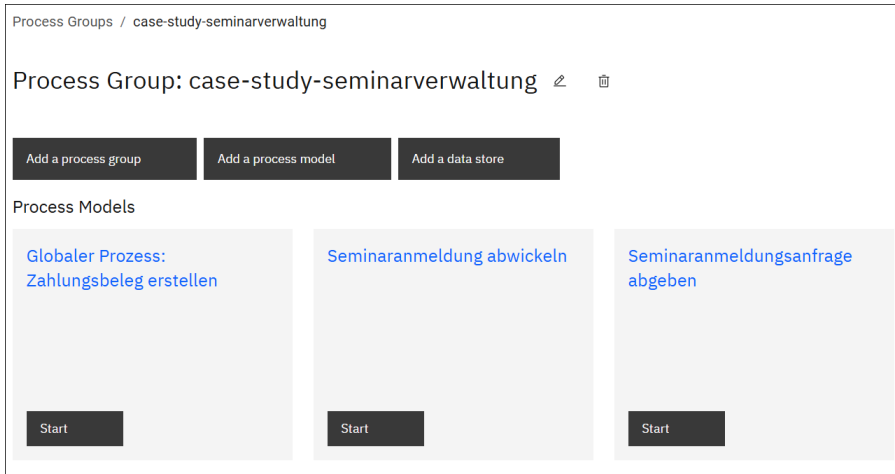


Abbildung 4.5: BPMN-Process-Engine »SpiffWorkflow« – Process Group »case-study-seminarverwaltung«

Damit haben Sie die erste Komponente der PDA-Seminarverwaltung aufgesetzt. Im nächsten Abschnitt finden Sie heraus, wie Sie die weiteren Komponenten aufsetzen.

4.4 Die weiteren Komponenten aufsetzen

Nachdem Sie die BPMN-Process-Engine *SpiffWorkflow* mit den Prozessmodellen installiert und eingerichtet haben, setzen wir nun die übrigen Komponenten ein: siehe dazu [Abbildung 1.1](#). Dazu finden Sie eine *docker-compose.yml*-Datei vor, sodass Sie diese Komponenten auch vereinfacht aufsetzen können. Gehen Sie dazu wie folgt vor:

1. Kopieren Sie aus dem entpackten Download-Ordner unter dem Kapitel 8-Verzeichnis folgende Ordner sowie die vorhandene *docker compose.yml*-Datei in einen beliebigen Ordner auf Ihrer Festplatte:
 - *PDA_SeminarAnmService*
 - *PDA_SemVerw-DecisionService*
 - *SCIL_Integrationskomponente*
2. Bitte entpacken Sie die Projektdateien in den Ordnern. Achten Sie dabei darauf, dass die Dateien direkt in diese Ordner entpackt werden und nicht in weitere Ordner verschachtelt sind. In [Abbildung 4.6](#) sehen Sie beispielhaft den Ordner *PDA_SemVerw-DecisionService*.

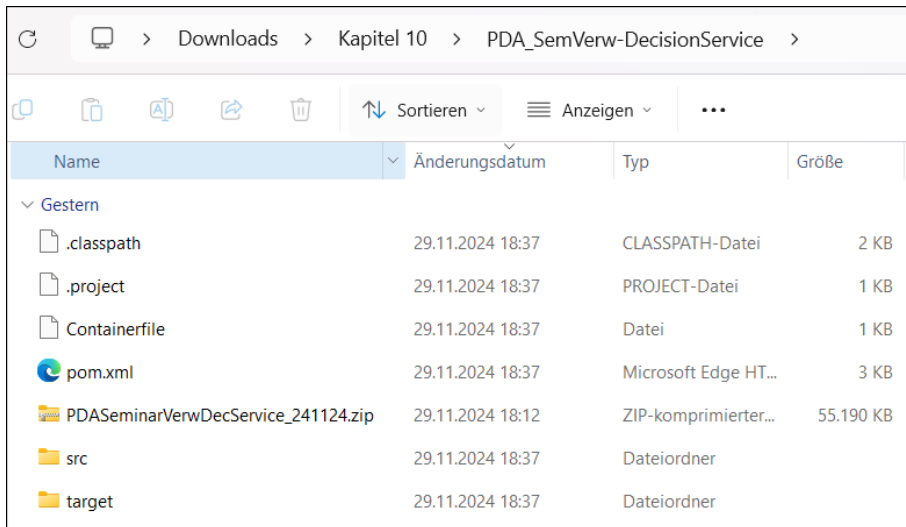


Abbildung 4.6: Weitere Komponenten der PDA »Seminarverwaltung« – Beispiel-Ordner zum Projekt »PDA_SemVerw-DecisionService«

- Die `docker-compose.yml` zu den Komponenten legen Sie im übergeordneten Ordner ab. Das Resultat der vorbereiteten Ordnerstruktur finden Sie in [Abbildung 4.7](#) veranschaulicht.

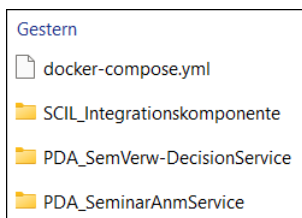


Abbildung 4.7: Weitere Komponenten der PDA-Seminarverwaltung – der übergeordnete Ordner mit allen weiteren Komponenten und der `docker-compose.yml`-Datei

- Klicken Sie in die obere Adressleiste der Ordneransicht, sodass der Ordnerpfad markiert ist. Kopieren Sie den Ordnerpfad in die Zwischenablage.
- Nun klicken Sie in die Windows-Suchleiste. Diese finden Sie in der Taskliste unten am Bildschirmrand. Tippen Sie »CMD« ein und öffnen Sie die Eingabeaufforderung, die in der Suchleiste erscheint.
- Geben Sie den folgenden Befehl ein: `cd`. Fügen Sie ein Leerzeichen und anschließend den Ordnerpfad aus Ihrer Zwischenablage ein. Dies können Sie mit der Tastenkombination `[Strg] + [V]` erledigen. Betätigen Sie anschließend die `[↵]`-Taste.

7. Öffnen Sie außerhalb der Eingabeaufforderung die *docker-compose.yml*-Datei in einem beliebigen Texteditor und scrollen Sie zu der Stelle mit den Umgebungsvariablen (environment) zu der Komponente *scilseminarverwintegrationskomponente*. In **Abbildung 4.8** ist die entsprechende Stelle in der Datei dargestellt.

```
scilseminarverwintegrationskomponente:
  build:
    context: .\SCIL_Integrationskomponente
    dockerfile: Containerfile
  container_name: scil-seminarverw-integrationskomponente
  ports:
    - "8083:8083"
  environment:
    - PROCESS_ENGINE_BACKEND_ENDPOINT_API_URL=http://localhost:8000/v1.0
    - PROCESS_ENGINE_BACKEND_API_KEY=122de0fe-e562-404e-a9b2-37b87e8ebe12
    - ACTIVE_MQ_BROKER_URL=tcp://host.docker.internal:61616
    - ACTIVE_MQ_BROKER_USER=admin
    - ACTIVE_MQ_BROKER_PWD=admin
```

Abbildung 4.8: Weitere Komponenten der PDA »Seminarverwaltung« – Prüfen und Anpassen der Umgebungsvariablen der SCIL-Integrationskomponente in der *docker-compose.yml*

Prüfen Sie die Werte der Umgebungsvariablen:

- Die Process-Engine-Backend-URL sollte korrekt sein. Der Port für das SpiffWorkflow-Backend darf nicht von der Angabe 8000 abweichen.
 - Den Wert der Umgebungsvariable `PROCESS_ENGINE_BACKEND_API_KEY` müssen Sie auf jeden Fall anpassen: Geben Sie nach dem Gleichheitszeichen direkt den erzeugten Backend-API-Key ein. Diesen haben Sie im Rahmen des Aufsetzens der BPMN-Process-Engine *SpiffWorkflow* erzeugt (siehe Abschnitt 4.3, »Die BPMN-Process-Engine »SpiffWorkflow« aufsetzen«). Anschließend speichern Sie die Änderungen der Datei ab, z. B. mit `[Strg] + [S]`.
8. Kehren Sie nun zur geöffneten Eingabeaufforderung zurück. Führen Sie folgende Befehle nacheinander aus:
- `podman compose build` `[↵]` – Damit werden die Images der eigenen Komponenten auf Basis der Containerfiles gebaut.
 - `docker-compose pull` `[↵]` Mit diesen beiden Befehlen wurden die Images gebaut und die benötigten Images aus dem Docker-Hub des Messaging-Systems *Apache ActiveMQ Classic* heruntergeladen.
 - `docker-compose up -d` `[↵]` – Damit starten Sie die Container zu den einzelnen Komponenten.
9. Nun können Sie unter dem Bereich **Container** in *Podman Desktop* die laufenden Container zur PDA *Seminarverwaltung* sehen.

4.5 Walkthrough durch die Prozessgesteuerte Anwendung

4.5.1 Voraussetzung

Die aufgesetzten Container der Komponenten laufen. Dies können Sie in *Podman Desktop* im Bereich **Containers** überprüfen. Sollten diese nicht laufen, klicken Sie auf das **Play**-Symbol bei den jeweiligen Komponenten.

4.5.2 Szenario I: Zum angefragten Seminar wird eine Rechnung zur Seminaranmeldung erzeugt

1. **Vorbereitung:** Im angemeldeten SpiffWorkflow-Frontend² klicken Sie auf **Processes** und öffnen das Prozessmodell *Globaler Prozess: Zahlungsbeleg erstellen* wie folgt:
 - Klicken Sie auf die Prozessgruppe **case-study-seminarverwaltung**.
 - Klicken Sie das Prozessmodell **Globaler Prozess: Zahlungsbeleg erstellen** an.
 - Klicken Sie die BPMN-Datei *ZahlungsbelegErstellen.bpmn* an.

Klicken Sie nun auf die Aufgabe **Zahlungssituation bestimmen** im Prozess. Anschließend klicken Sie im rechten Property-Bereich auf den Tab **Script**. Stellen Sie sicher, dass das Skript wie folgt aussieht:

```
zahlungsSituation="INVOICE"
```

Um die Änderung im BPMN-Diagramm zu speichern, klicken Sie auf den Button **Save**.

2. Navigieren Sie über den obigen Navigationspfad zurück zum Einstieg der Prozessgruppe **case-study-seminarverwaltung**. Klicken Sie dazu im Navigationspfad **case-study-seminarverwaltung** an. Nun sehen Sie die Übersicht der Prozessmodelle dieser Prozessgruppe.
3. Klicken Sie auf den Button **Start** des Prozessmodells *Seminaranmeldungsanfrage abgeben*.
4. Sie werden direkt zur User Task Form *Seminaranmeldungsabfrage erfassen* geleitet. Füllen Sie die erforderlichen Felder aus. Abschließend klicken Sie auf den Button **Submit**.
5. Sie sehen nun die abgelaufene Prozessinstanz zum Fachprozess *Seminaranmeldungsanfrage abgeben* mit dem Status **complete**.
6. Sie können den zur Laufzeit ausgewählten Pfad der Prozessinstanz inspizieren. Zudem können Sie auf die einzelnen BPMN-Elemente klicken, um die dort jeweils zur Laufzeit vorhandenen Taskdaten zu inspizieren. Ferner können Sie auf das Datenobjekt **seminaranmeldungsanfrage** klicken, um den Datenzustand dieses Objekts anzuzeigen. Die Ansicht können Sie in Abbildung 4.9 exemplarisch sehen.

2 Geben Sie im Browser die URL *http://localhost:8001/* ein und melden Sie sich wie folgt an:
Benutzer: »admin«, Passwort: »admin«.

Um die Datenansicht des Datenobjekts zu schließen, klicken Sie auf eine Stelle neben dem Fenster der Datenansicht.

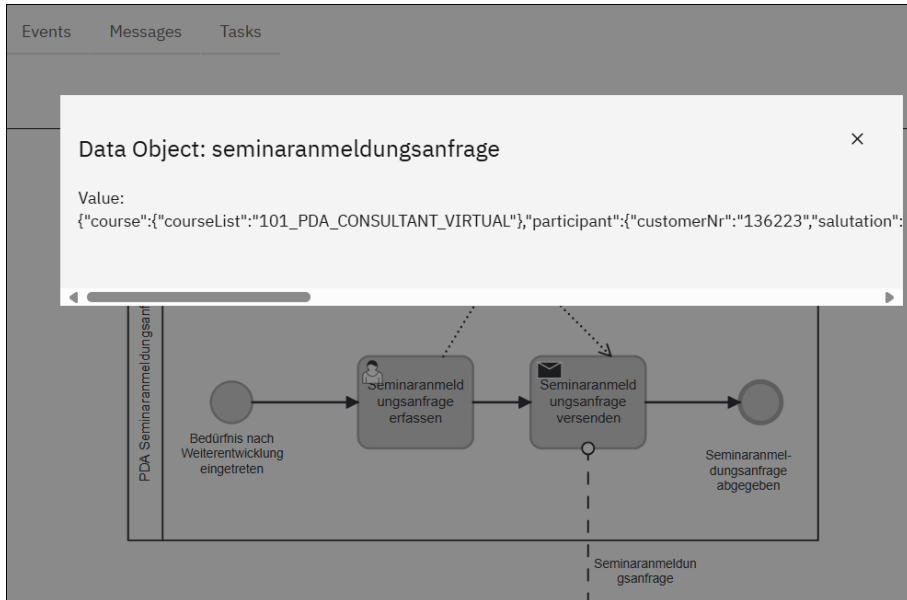


Abbildung 4.9: Walkthrough – Prozess »Seminaranmeldungsanfrage abgeben«: Ansicht des Datenobjekts »Seminaranmeldungsanfrage« und des abgelaufenen Pfades der Prozessinstanz

7. Klicken Sie nun auf das Menü **Process Instances** im oberen Bereich. Hier können Sie sehen, dass eine **Process Instance »Seminaranmeldung abwickeln«** gestartet wurde. Klicken Sie auf den Eintrag zur Prozessinstanz.
8. Sie kommen zur Ansicht der Prozessinstanz (siehe [Abbildung 4.10](#)). Unter **Tasks I can complete** finden Sie die User Tasks, die Sie bearbeiten können. Weiter unten finden Sie die Ansicht des Fachprozesses mit dem aktuellen Stand der Prozessausführung. Per Klick auf die Call Activity *Seminaranmeldung verbuchen* können Sie sich einen genaueren Einblick von dem aktuellen Zustand verschaffen. Die gelbe Einfärbung zeigt Ihnen, wo sich die Ausführung aktuell im Fachprozess befindet. Auch können Sie sich die aktuellen Prozessdaten anschauen. Dazu gehen Sie wie im vorherigen Schritt beschrieben vor.

SpiffWorkflow localhost A

Tasks I can complete

Task	Waiting for	Actions
Seminaranmeldung speichern	admin@example.com	<button>Go</button>

Diagram Milestones Events Messages Tasks

Abbildung 4.10: Walkthrough – Prozess »Seminaranmeldung abwickeln« wartet zur Laufzeit, dass der Benutzer die Aufgabe »Seminaranmeldung speichern« erledigt.

Klicken Sie nun auf den Button **Go** bei der Aufgabe **Seminaranmeldung speichern** im Abschnitt **Tasks I can complete** (siehe [Abbildung 4.10](#)).

- Daraufhin wird Ihnen eine neue Seite angezeigt mit der Überschrift **Task: Seminaranmeldung speichern (Seminaranmeldung abwickeln)**. Klicken Sie auf den Button **Submit** (siehe [Abbildung 4.11](#)).

Diese Aufgabe ist eine undefinierte BPMN-Aufgabe im Fachprozess. SpiffWorkflow interpretiert aktuell undefinierte Aufgaben wie eine Art User Task.

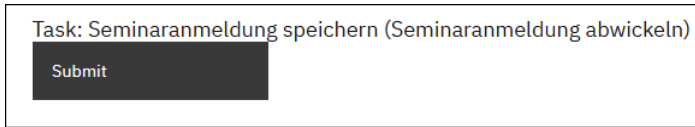


Abbildung 4.11: Walkthrough – Prozess »Seminaranmeldung abwickeln«:
Darstellung der Aufgabe »Seminaranmeldung speichern« im Frontend

10. Der für aus End-User-Sicht nächste sichtbare Schritt ist die undefinierte Aufgabe *Rechnung erstellen*. Wenn Sie hier angelangt sind, wurde vorher der zur Prozessgesteuerten Anwendung gehörende *PDASeminarverwaltung-DecisionService* abgefragt. Hier verbirgt sich das DMN-Entscheidungsmodell *Stornierungsgebühren ermitteln*, das im Rahmen des *Kogito-Decision-Service* evaluiert wird. Dieses Entscheidungsmodell finden Sie in den Downloadmaterialien unter *Kapitel 5-6 • PDA Seminarverwaltung • 02_DMN_Entscheidungsmodell*.

Wenn Sie das Entscheidungsmodell *Stornierungsgebühren ermitteln* komfortabel evaluieren wollen, können Sie über die *Swagger-UI* des *Decision-Service* das Entscheidungsmodell ausführen. Dazu geben Sie in einem neuen Tab Ihres Browsers die folgende URL ein: <http://localhost:8082/swagger-ui/index.html>. Die möglichen Input-Informationen für das Entscheidungsmodell entnehmen Sie bitte dem DMN-Entscheidungsmodell in den Downloadmaterialien.

Klicken Sie bei der Aufgabe **Task: Rechnung erstellen (Seminaranmeldung abwickeln)** auf den Button **Submit**.

11. Um die Ausführung des Fachprozesses *Seminaranmeldung abwickeln* zu beenden, folgen nun weitere Aufgaben, die Sie jeweils mit dem Button **Submit** abschließen. Dies sind folgende Aufgaben, die ich im Rahmen dieses Buches aber nicht umgesetzt habe:
- Task: Bestätigung Seminaranmeldung versenden (Seminaranmeldung abwickeln),
 - Task: Teilnahmebescheinigung/Zertifikat erstellen (Seminaranmeldung abwickeln),
 - Task: Seminaranmeldung ändern (Seminaranmeldung abwickeln)

Damit ist die Prozessinstanz beendet. Ihnen wird die Ansicht **Process Instance** angezeigt (siehe [Abbildung 4.12](#)). Wenn Sie Lust haben, können Sie den ausgeführten Pfad über die Markierung im dargestellten BPMN-Fachprozess ansehen oder/und die Prozessdaten inspizieren.

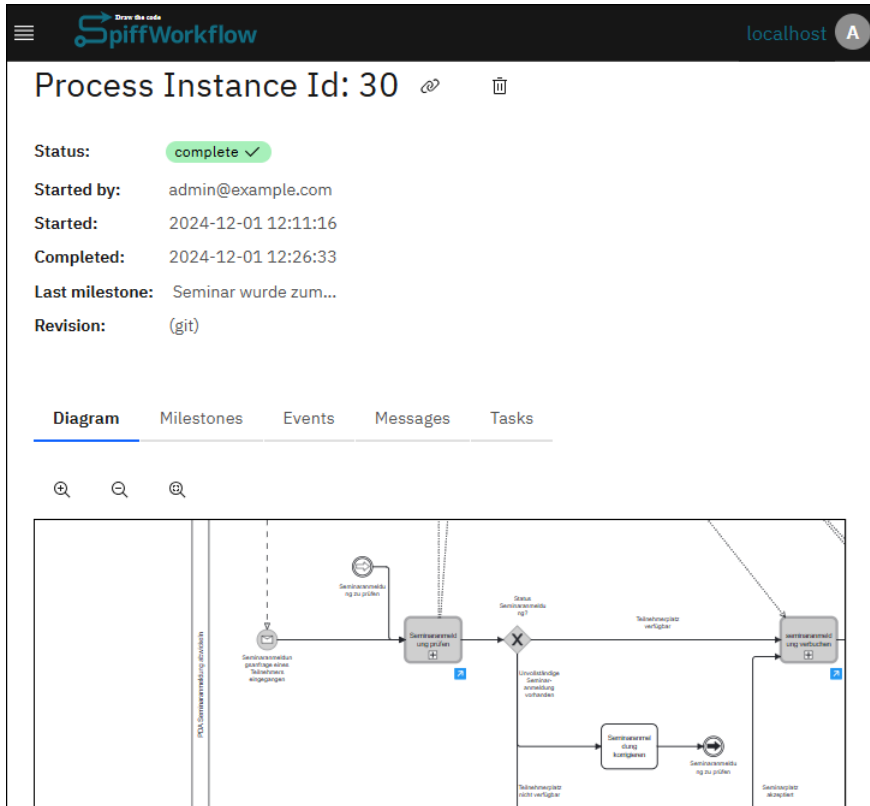


Abbildung 4.12: Walkthrough – Prozess »Seminaranmeldung abwickeln«: Darstellung der abgeschlossenen Prozessinstanz (Auszug)

4.5.3 Szenario II: Zum angefragten Seminar wird eine Gutschrift zur Seminaranmeldung erzeugt

1. **Vorbereitung:** Im angemeldeten SpiffWorkflow-Frontend³ klicken Sie auf **Processes** und öffnen das Prozessmodell *Globaler Prozess: Zahlungsbeleg erstellen* wie folgt:
 - Klicken Sie auf die Prozessgruppe **case-study-seminarverwaltung**.
 - Klicken Sie auf das Prozessmodell **Globaler Prozess: Zahlungsbeleg erstellen**.
 - Klicken Sie auf die BPMN-Datei *ZahlungsbelegErstellen.bpmn*.

Klicken Sie nun auf die Aufgabe **Zahlungssituation bestimmen** im Prozess. Anschließend klicken Sie im rechten Property-Bereich auf den Tab **Script**. Stellen Sie sicher, dass das Skript wie folgt aussieht:

```
zahlungsSituation="VOUCHER"
```

Um die Änderung im BPMN-Diagramm zu speichern, klicken Sie auf den Button **Save**.

³ Geben Sie im Browser die URL <http://localhost:8001/> ein und melden Sie sich wie folgt an:
Benutzer: »admin«, Passwort: »admin«.

2. Navigieren Sie über den obigen Navigationspfad zurück zum Einstieg der Prozessgruppe **case-study-seminarverwaltung**, indem Sie auf sie klicken. Nun sehen Sie die Übersicht der Prozessmodelle dieser Prozessgruppe.
3. Klicken Sie auf den Button **Start** des Prozessmodells *Seminaranmeldungsanfrage abgeben*.
4. Sie werden direkt zur User Task Form **Seminaranmeldungsabfrage erfassen** geleitet. Füllen Sie die erforderlichen Felder aus. Abschließend klicken Sie auf den Button **Submit**.
5. Sie sehen nun die abgelaufene Prozessinstanz zum Fachprozess *Seminaranmeldungsanfrage abgeben* mit dem Status **complete**.
6. Nun können Sie den zur Laufzeit ausgewählten Pfad der Prozessinstanz oder die zur Laufzeit vorhandenen Taskdaten der einzelnen BPMN-Elemente inspizieren. Ferner können Sie auf das Datenobjekt **seminaranmeldungsanfrage** klicken, um den Datenzustand dieses Objekts anzuzeigen. Diese Ansicht sehen Sie in Abbildung 4.13.

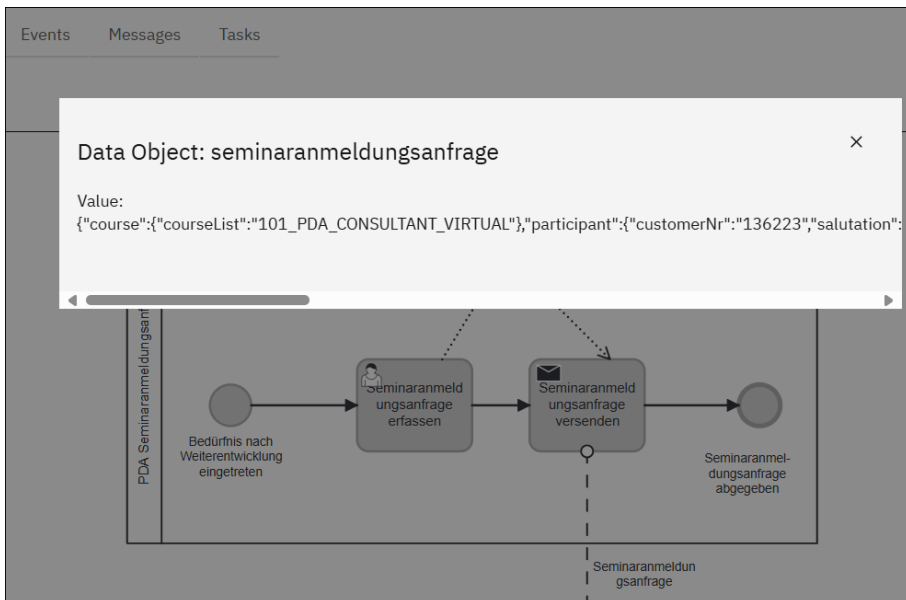


Abbildung 4.13: Walkthrough – Prozess »Seminaranmeldungsanfrage abgeben«: Ansicht des Datenobjekts »Seminaranmeldungsanfrage« und des abgelaufenen Pfades der Prozessinstanz

7. Klicken Sie nun im oberen Menü auf **Process Instances**. Eine Prozessinstanz *Seminaranmeldung abwickeln* wurde gestartet. Klicken Sie auf den Eintrag zur Prozessinstanz.
8. Sie kommen zur Ansicht der Prozessinstanz (siehe Abbildung 4.14). Unter **Tasks I can complete** finden Sie die User Tasks, die Sie bearbeiten können. Weiter unten finden Sie

die Ansicht des Fachprozesses mit dem aktuellen Stand der Prozessausführung. Durch einen Klick auf die Call Activity *Seminaranmeldung verbuchen* können Sie sich einen genaueren Überblick über den aktuellen Zustand verschaffen. Die gelbe Einfärbung zeigt Ihnen, wo sich die Ausführung aktuell im Fachprozess befindet. Auch können Sie sich die aktuellen Prozessdaten anschauen. Dazu gehen Sie so vor, wie im vorherigen Schritt beschrieben.

The screenshot shows the SpiffWorkflow application interface. At the top, there is a header with the SpiffWorkflow logo and a 'localhost' indicator. Below the header, the main content area is titled 'Tasks I can complete'. It contains a table with the following data:

Task	Waiting for	Actions
Seminaranmeldung speichern	admin@example.com	<button>Go</button>

Below the table, there are tabs for 'Diagram', 'Milestones', 'Events', 'Messages', and 'Tasks'. The 'Diagram' tab is selected, showing a BPMN diagram. The diagram illustrates a process flow with various activities, including a yellow-highlighted activity labeled 'Seminaranmeldung speichern' (12). The diagram also shows message flows and data stores.

Abbildung 4.14: Walkthrough – Prozess »Seminaranmeldung abwickeln« wartet zur Laufzeit, dass der Benutzer die Aufgabe »Seminaranmeldung speichern« erledigt.

Klicken Sie nun auf den Button **Go** bei der Aufgabe **Seminaranmeldung speichern** im Abschnitt **Tasks I can complete** (siehe [Abbildung 4.14](#)).

- Daraufhin wird Ihnen eine neue Seite mit der Überschrift *Task: Seminaranmeldung speichern* (Seminaranmeldung abwickeln) angezeigt. Klicken Sie hier auf den Button **Submit** (siehe [Abbildung 4.15](#)).

Diese Aufgabe ist eine undefinierte BPMN-Aufgabe im Fachprozess. SpiffWorkflow interpretiert aktuell undefinierte Aufgaben wie eine Art User Task.

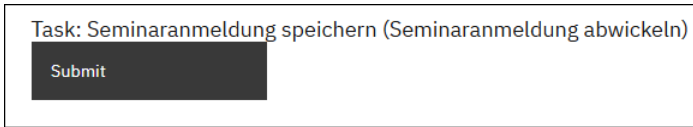


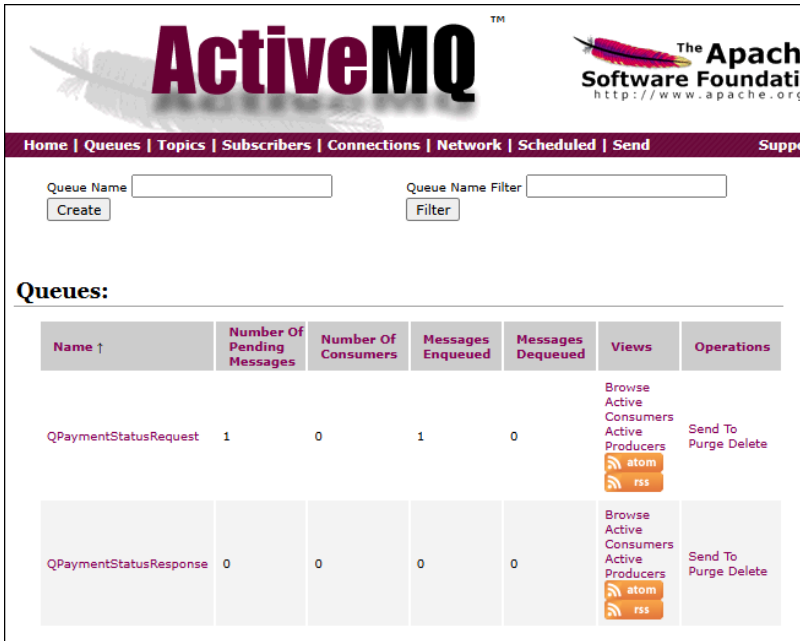
Abbildung 4.15: Walkthrough – Prozess »Seminaranmeldung abwickeln«;
Darstellung der Aufgabe »Seminaranmeldung speichern« im Frontend

Nun gelangen Sie innerhalb des globalen Fachprozesses *Zahlungsbeleg erstellen* in den Alternativpfad *Zahlungssituation? – Gutschrift*. Die Ausführung der Schritte erfolgt sehr schnell. Wenn Sie das Zusammenspiel der Komponenten zwischen *PDA Seminarverwaltung*, dem Messaging-System *ActiveMQ Classic* sowie der *SCIL-Integrationskomponente* besser überblicken möchten, empfehle ich Ihnen, den Container der SCIL-Integrationskomponente über das zugehörige Symbol **Stop Container** in Podman zu stoppen. Später können Sie den Container wieder starten.

10. Wenn Sie den Container der SCIL-Integrationskomponente gestoppt haben, können Sie nun im Messaging-System innerhalb der Queue *QPaymentStatusRequest* die relevanten übergebenen Informationen zur Anforderung der vorhandenen Funktionalität *Bisherigen Zahlungsstand ermitteln* inspizieren. Ansonsten fahren Sie direkt mit Schritt 13 fort, um die Prozessinstanz zu beenden.

Zum Inspizieren der Queue innerhalb des Messaging-Systems verwenden Sie die webbasierte Oberfläche:


- Dazu tippen Sie in einem neuen Tab Ihres Browsers folgende URL ein: *http://localhost:8161/*.
- Auf der Startseite klicken Sie auf den Link **Manage ActiveMQ Broker**.
- Nun müssen Sie zur Autorisierung einen Benutzernamen und ein Passwort eingeben. Sie geben als Benutzer »admin« und als zugehöriges Passwort »admin« ein.
- Danach haben Sie Zugriff auf die *ActiveMQ Console*. Klicken Sie in der oberen Navigationsleiste auf **Queues**.
- Sie sehen nun die vorhandenen Queues. Wenn die SCIL-Integrationskomponente bereits gestartet ist, werden wie in Abbildung 4.16 die benötigten Queues angelegt sein. In der Queue **QPaymentStatusRequest** finden Sie die übermittelten fachlichen Daten aus der PDA, um die Funktionalität *Bisherigen Zahlungsstand ermitteln* aus der Systemlandschaft anzufordern. Klicken Sie auf den Namen der Queue **QPaymentStatusRequest**, um eine Übersicht über die aktuell vorhandenen Messages zu erhalten (siehe Abbildung 4.17).



The screenshot shows the ActiveMQ web console interface. At the top, there is a navigation bar with links: Home, Queues, Topics, Subscribers, Connections, Network, Scheduled, Send, and Support. Below the navigation bar, there are input fields for 'Queue Name' and 'Queue Name Filter', each with a 'Create' or 'Filter' button. The main section is titled 'Queues:' and contains a table with the following columns: Name, Number Of Pending Messages, Number Of Consumers, Messages Enqueued, Messages Dequeued, Views, and Operations.

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
QPaymentStatusRequest	1	0	1	0	Browse Active Consumers Active Producers atom RSS	Send To Purge Delete
QPaymentStatusResponse	0	0	0	0	Browse Active Consumers Active Producers atom RSS	Send To Purge Delete

Abbildung 4.16: Walkthrough – Übersicht über die Queues zur Anforderung der vorhandenen Funktionalität »Bisherigen Zahlungsstand ermitteln«



The screenshot shows the 'Browse QPaymentStatusRequest' page in the ActiveMQ web console. It displays a table with message details. The table has columns: Message ID, Correlation ID, Persistence, Priority, Redelivered, Reply To, Timestamp, Type, and Operations.

Message ID	Correlation ID	Persistence	Priority	Redelivered	Reply To	Timestamp	Type	Operations
ID:38c1bb5ec779-43809-1733051201392-4:1:1:1:1		Persistent	5	false		2024-12-01 12:08:07:661 UTC		Delete

Below the table, there is a link 'View Consumers'.

Abbildung 4.17: Walkthrough – Übersicht über die Messages innerhalb der Queue »QPaymentStatusRequest«

- Klicken Sie auf die vorhandene Message-ID, um zu den Details der Message zu gelangen. Unter dem Abschnitt **Message Details** können Sie auch die Inhalte der Message sehen (siehe auch [Abbildung 4.18](#)), z. B.: {"seminarRegNr": "UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHz"}

Message ID	ID:38c1bb5ec779-43809-17330512013
Destination	queue://QPaymentStatusRequest
Correlation ID	
Group	
Sequence	0
Expiration	0
Persistence	Persistent
Priority	5
Redelivered	false
Reply To	
Timestamp	2024-12-01 12:08:07:661 UTC
Type	

Message Actions

Delete

Copy

Move

-- Please select --

Message Details

{"seminarRegNr": "UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHvz"}

Abbildung 4.18: Walkthrough – Ansicht einer Message innerhalb der Queue »QPayment-StatusRequest« zur Anforderung der vorhandenen Funktionalität »Bisherigen Zahlungsstand ermitteln« (Auszug)

11. Schauen Sie sich im Fachprozessmodell der Prozessinstanz *Seminaranmeldung abwickeln* die Aufgabe *Anfrage »Bisherigen Zahlungsstand ermitteln«* senden an. Diese finden Sie innerhalb der Call Activities *Seminaranmeldung verbuchen* und *Zahlungsbeleg erstellen*. Dort finden Sie in den Taskdaten der Aufgabe den Hinweis, dass via HTTP-Request die erforderlichen fachlichen Informationen als JSON-Struktur an das Messaging-System gesendet wurden:

```
"sendReqBishZahlungsstandErmResp": {
  "body": {
    "raw_response": "Message sent"
  },
  "mimetype": "application/json",
  "http_status": 200,
  "operator_identifier": "http/PostRequestV2"
}
```


12. Als Nächstes muss die SCIL-Integrationskomponente diese Message aus der Queue **QPaymentStatusRequest** abholen und so verarbeiten, dass diese für den simulierten Payment-Service-Aufruf verwendet werden kann. Anschließend gibt sie die erhaltene Antwort in der für den PDA-Fachprozess relevanten fachlichen Information zurück. Dazu wird sie die Daten mit der Process-Engine-Backend-spezifischen API als BPMN-Message senden. Diese Information wird von der Receive Task *Antwort »Bisherigen Zahlungsstand ermitteln« empfangen* aufgegriffen. Danach geht der Fachprozess gemäß BPMN-Prozessmodell weiter.

Damit dieser Ablauf durchgeführt werden kann, starten Sie den Container der SCIL-Integrationskomponente in Podman über das zugehörige **Play**-Symbol.

13. Damit Sie den Ablauf in groben Zügen nachvollziehen können, protokolliert die SCIL-Integrationskomponente wesentliche Schritte. Klicken Sie in Podman auf ... beim laufenden Container der SCIL-Integrationskomponente. Im dann erscheinenden Kontextmenü klicken Sie auf den Eintrag **Open Logs**.

Nun sehen Sie folgende interessante Stellen:

```
Payment Status Service Request for Seminar registration number:
UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHz
```

Hier wird dokumentiert, dass die Payment-Status-Anfrage aus der Messaging-Queue von der SCIL-Integrationskomponente abgerufen wurde.

```
2024-12-01 13:26:44.422 INFO 1 --- [StatusResponse]] route2
: Process Engine Request Body Trigger Message: {"seminarRegNr":
"UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHz", "paymentStatus": {"date":
"2024-12-01", "amount": "1500.50", "currency": "EUR"}}
```

Die von der PDA-erwartete Antwortstruktur wird als BPMN-Message versendet, so dass die Aufgabe *Antwort »Bisherigen Zahlungsstand ermitteln« empfangen* innerhalb des Fachprozesses die Informationen empfangen kann. Dazu wird die Process-Engine-Backend-spezifische API⁴ verwendet.

```
2024-12-01 13:26:44.796 INFO 1 --- [StatusResponse]] route2
: Process Engine Response 'Trigger Message': {"task_data": {"zahlungsbe-
legInput": {"seminarRegNr": "UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHz",
"seminarBeginn": "2024-11-17", "hasBeenCancelled": false, "cancelIni-
tiator": ""}, "course": {"price": 1500.5}, "zahlungsSituation": "VOU-
CHER", "sendReqBishZahlungsstandErmResp": {"body": {"raw_response":
"Message sent"}, "mimetype": "application/json", "http_status": 200,
"operator_identifizier": "http/PostRequestV2"}, "recBishZahlungsstand-
Erm": {"seminarRegNr": "UwK5LZDW-USIm-Rj1I-andY-USM0RGkcFHz", "pay-
mentStatus": {"date": "2024-12-01", "amount": "1500.50", "currency":
```

⁴ Die *Swagger UI* der Process-Engine-Backend-API von SpiffWorkflow finden Sie hier unter der folgenden URL: <http://localhost:8000/v1.0/ui/>

```
"EUR"}}}, "process_instance": {"id": 34, "process_model_identifizier":  
"case-study-seminarverwaltung/seminaranmeldung-abwickeln", "process_  
model_display_name": "Seminaranmeldung abwickeln", "process_initi-  
ator_id": 1, "start_in_seconds": 1733059597, "end_in_seconds": null,  
"updated_at_in_seconds": 1733059605, "created_at_in_seconds":  
1733059597, "status": "user_input_required", "bpmn_version_control_  
identifizier": null}}"operator_identifizier": "http/PostRequestV2"},  
"recBishZahlungsstandErm": {"seminarRegNr": "4xUCAi8S-Rt3V-KNca-  
hBvx-eEqA6fafYkWG", "paymentStatus": {"date": "2024-12-01", "amount":  
"1500.50", "currency": "EUR"}}}, "process_instance": {"id": 34, "pro-  
cess_model_identifizier": "case-study-seminarverwaltung/seminaranmel-  
dung-abwickeln", "process_model_display_name": "Seminaranmeldung ab-  
wickeln", "process_initiator_id": 1, "start_in_seconds": 1733059597,  
"end_in_seconds": null, "updated_at_in_seconds": 1733059605, "created_  
at_in_seconds": 1733059597, "status": "user_input_required", "bpmn_  
version_control_identifizier": null}}
```

Hier finden Sie die protokollierte Process-Engine-Antwort, die vom Aufruf der vorherigen Anfrage kommt.

14. Kehren Sie zurück zur Prozessinstanz-Ansicht im SpiffWorkflow-Frontend. Dort sehen Sie gegebenenfalls nach Aktualisierung der Ansicht, dass die Prozessinstanz bei der Aufgabe *Gutschrift erstellen* angelangt ist.

Wenn Sie auf die Aufgabe **Antwort »Bisherigen Zahlungsstand ermitteln« empfangen** klicken, dann können Sie in den Prozessdaten der Aufgabe folgende Passage sehen:

```
"recBishZahlungsstandErm": {  
  "seminarRegNr": "4xUCAi8S-Rt3V-KNca-hBvx-eEqA6fafYkWG",  
  "paymentStatus": {  
    "date": "2024-12-01",  
    "amount": "1500.50",  
    "currency": "EUR"  
  }  
}
```

Dies sind die Daten, die von der SCIL-Integrationskomponente gemäß Servicevertrag der PDA übermittelt wurden.

15. Um die Ausführung der Prozessinstanz zu beenden, müssen Sie folgende Aufgaben noch per Klick auf den jeweiligen Button **Submit** im SpiffWorkflow-Frontend abschließen.

Zum Fortführen klicken Sie unter dem Abschnitt **Tasks I can complete** auf den Button **Go** zur Aufgabe **Gutschrift erstellen**. Anschließend müssen Sie folgende weitere Aufgaben durch Klick auf den Button **Submit** abarbeiten:

- Task: Bestätigung Seminaranmeldung versenden (Seminaranmeldung abwickeln)
- Task: Teilnahmebescheinigung/Zertifikat erstellen (Seminaranmeldung abwickeln)
- Task: Seminaranmeldung ändern (Seminaranmeldung abwickeln)

16. Damit ist die Prozessinstanz beendet. Ihnen wird die Ansicht **Process Instance** angezeigt (siehe [Abbildung 4.19](#)). Wenn Sie Lust haben, können Sie den ausgeführten Pfad über die Markierung im dargestellten BPMN-Fachprozess ansehen oder/und die Prozessdaten inspizieren.

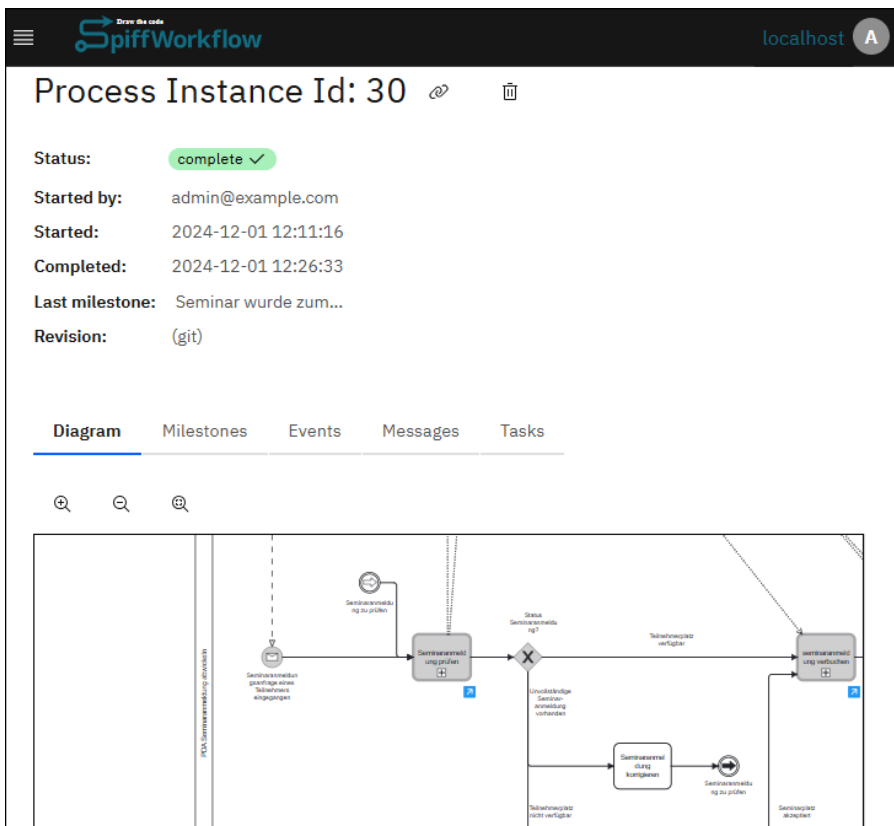


Abbildung 4.19: Walkthrough – Prozess »Seminaranmeldung abwickeln«; Darstellung der abgeschlossenen Prozessinstanz (Auszug und Beispiel)