Crear un flujo de trabajo en Zeebe Modeler

¿Eres nuevo en BPMN y quieres aprender más antes de seguir adelante? Esta publicación de blog ayuda a explicar el estándar y por qué es una buena opción para la orquestación de microservicios.

En caso de que ya esté familiarizado con BPMN y cómo crear un modelo BPMN en Zeebe Modeler, puede encontrar el modelo terminado que creamos durante el tutorial aquí: Zeebe Tutorial de inicio Modelo de flujo de trabajo tutorial .

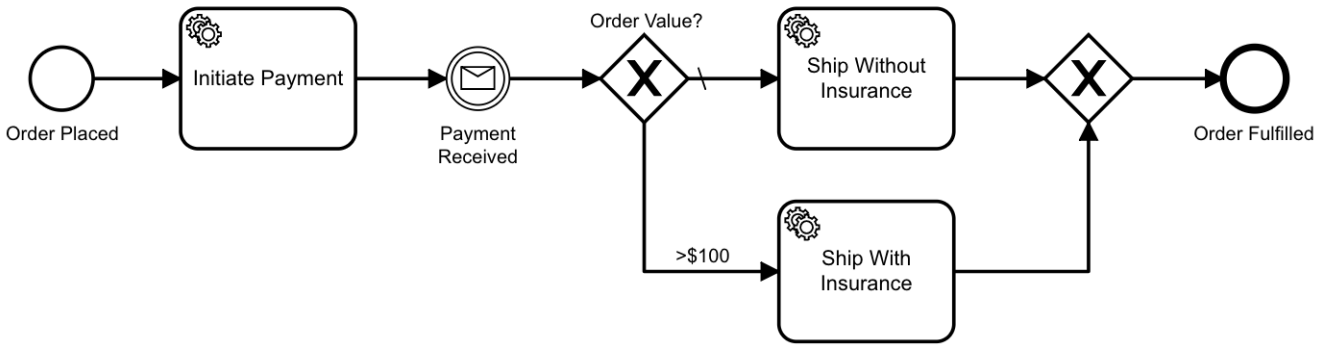
Si está utilizando el modelo terminado que proporcionamos en lugar de construir el suyo propio, también puede pasar a la sección 3.3: Implementar un flujo de trabajo .

Zeebe Modeler es una herramienta de modelado de escritorio que le permite construir y configurar modelos de flujo de trabajo utilizando BPMN 2.0. En esta sección, crearemos un modelo de flujo de trabajo y lo prepararemos para implementarlo en Zeebe.

Crearemos un proceso de pedido de comercio electrónico como nuestro ejemplo, y modelaremos un flujo de trabajo que consiste en:

- Iniciar un pago por un pedido
- Recibir un mensaje de confirmación de pago de un sistema externo
- Envío de los artículos en el pedido con o sin seguro dependiendo del valor del pedido

Así será su modelo de flujo de trabajo cuando terminemos:



La tarea de pago y las tareas de envío se llevan a cabo mediante servicios de trabajadores que conectaremos al motor de flujo de trabajo. El mensaje "Pago recibido" será publicado en Zeebe por un sistema externo, y Zeebe correlacionará el mensaje con una instancia de flujo de trabajo.

Para empezar

- Abra el Modelador Zeebe y cree un nuevo diagrama BPMN.
- Guarde el modelo como `order-process.bpmn` en el nivel superior del directorio de intermediarios de Zeebe que acaba de descargar. Como recordatorio, este directorio se llama `zeebe-broker-0.17.0`

El primer elemento en su modelo será un Evento de inicio, que ya debería estar en el lienzo cuando abra el Modelador.

Es una práctica recomendada de BPMN etiquetar todos los elementos de nuestro modelo, por lo tanto:

- Haga doble clic en el evento de inicio
- Rotúlelo "Pedido realizado" para indicar que nuestro proceso se iniciará siempre que un cliente haga un pedido

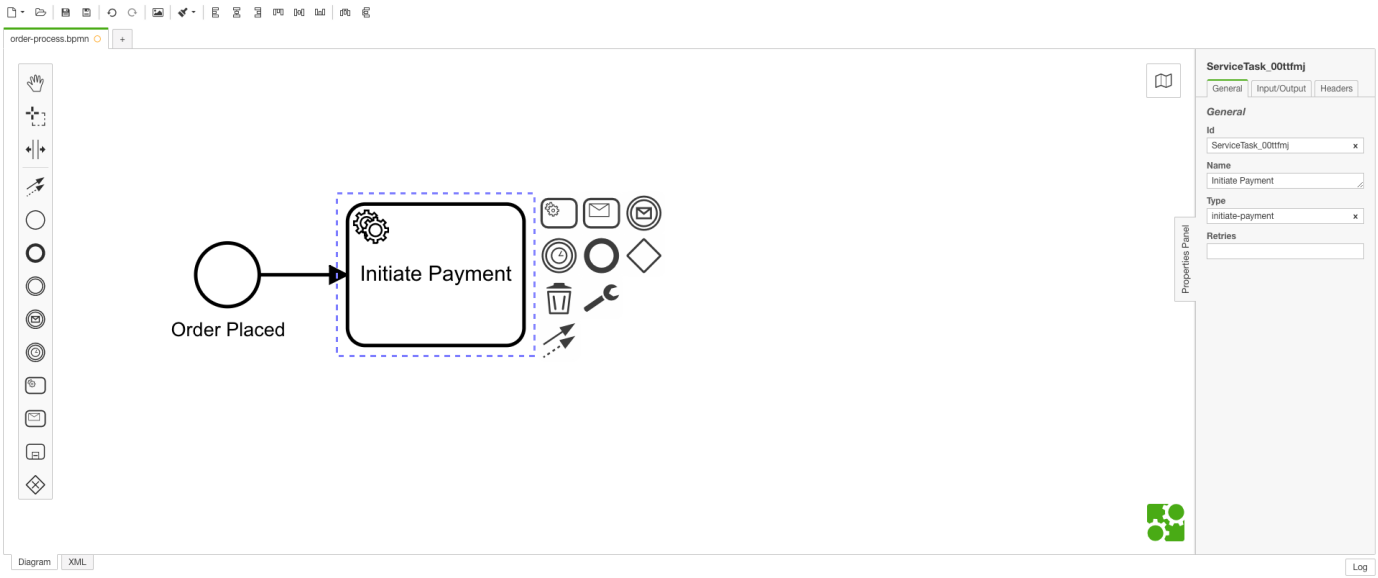
A continuación, debemos agregar una Tarea de servicio:

- Haga clic en el evento de inicio y seleccione el ícono Tarea de servicio
- Etiquete la Tarea de servicio "Iniciar pago"

A continuación, configuraremos la tarea de servicio "Iniciar pago" para que un microservicio externo pueda trabajar en ella:

- Haga clic en la tarea "Iniciar pago"
- Expanda el panel Propiedades en el lado derecho de la pantalla si aún no está visible
- En el campo *Tipo* en el panel Propiedades, ingrese `initiate-payment`

Esto es lo que debería ver en su Modelador ahora.



Este campo *Tipo* representa el *tipo de trabajo* en Zeebe. Un par de conceptos que es importante entender en este punto:

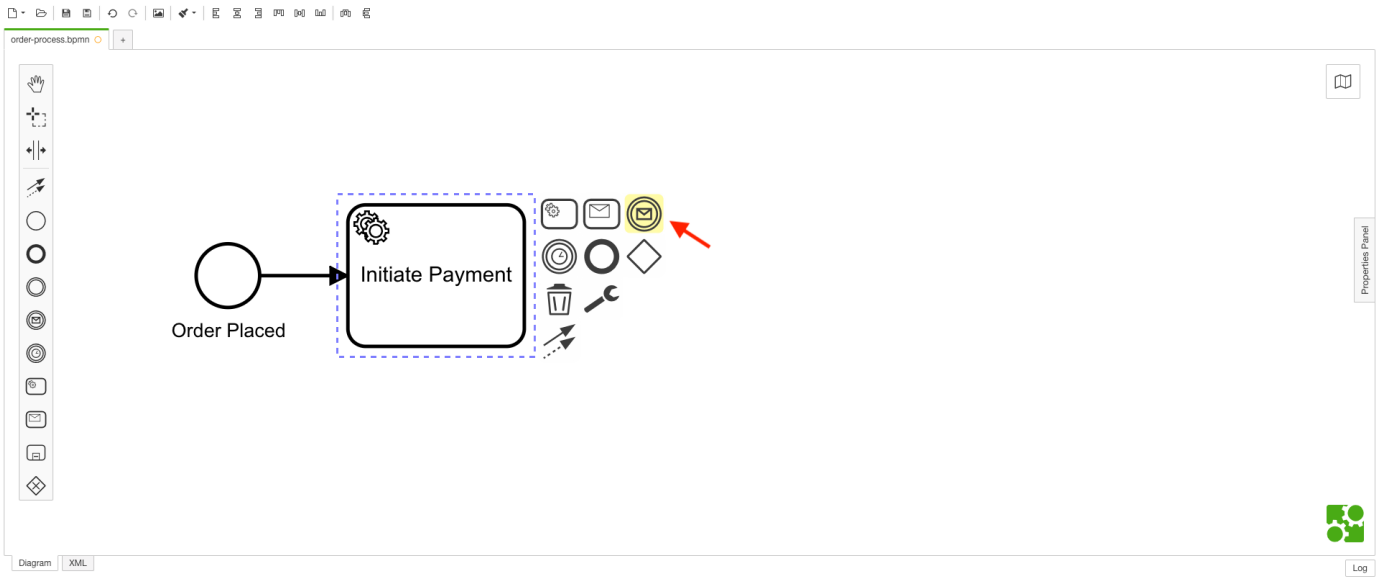
- A *job* is simply a work item in a workflow that needs to be completed before a workflow instance can proceed to the next step. (*See: Job Workers*)
- A *workflow instance* is one running instance of a workflow model--in our case, an individual order to be fulfilled. (*See: Workflows*)

For every workflow instance that arrives at the "Initiate Payment" Service Task, Zeebe will create a job with type `initiate-payment`. The external worker service responsible for payment processing--the so-called job worker--will poll Zeebe intermittently to ask if any jobs of type `initiate-payment` are available.

If a job is available for a given workflow instance, the worker will activate it, complete it, and notify Zeebe. Zeebe will then advance that workflow instance to the next step in the workflow.

Next, we'll add a Message Event to the workflow:

- Click on the "Initiate Payment" task on the Modeler
- Select the circular icon with an envelope in the middle
- Double-click on the message event and label it "Payment Received"

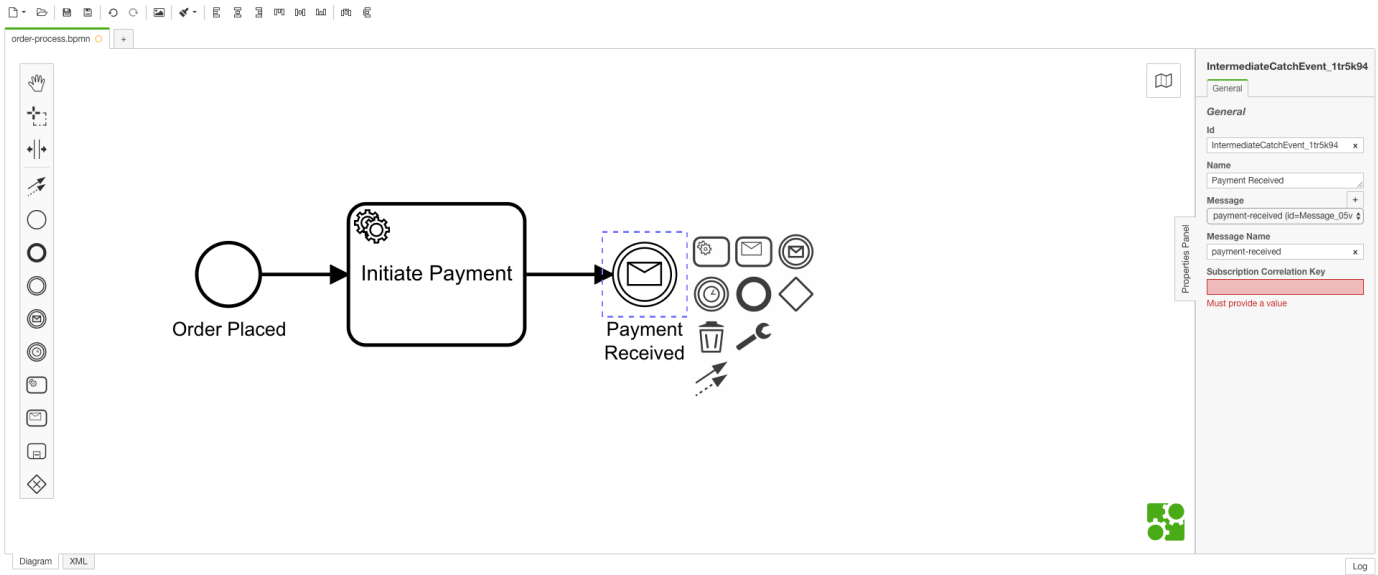


We use message catch events in Zeebe when the workflow engine needs to receive a message from an external system before the workflow instance can advance. (*See: Message Events*)

In the scenario we're modeling, we *initiate* a payment with our Service Task, but we need to wait for some other external system to actually confirm that the payment was received. This confirmation comes in the form of a message that will be sent to Zeebe - asynchronously - by an external service.

Messages received by Zeebe need to be correlated to specific workflow instances. To make this possible, we have some more configuring to do:

- Select the Message Event and make sure you're on the "General" tab of the Properties panel on the right side of the screen
- In the Properties panel, click the `+` icon to create a new message. You'll now see two fields in the Modeler that we'll use to correlate a message to a specific workflow instance: Message Name and Subscription Correlation Key.
- Let's give this message a self-explanatory name: `payment-received`.



When Zeebe receives a message, this name field lets us know *which message event in the workflow model* the message is referring to.

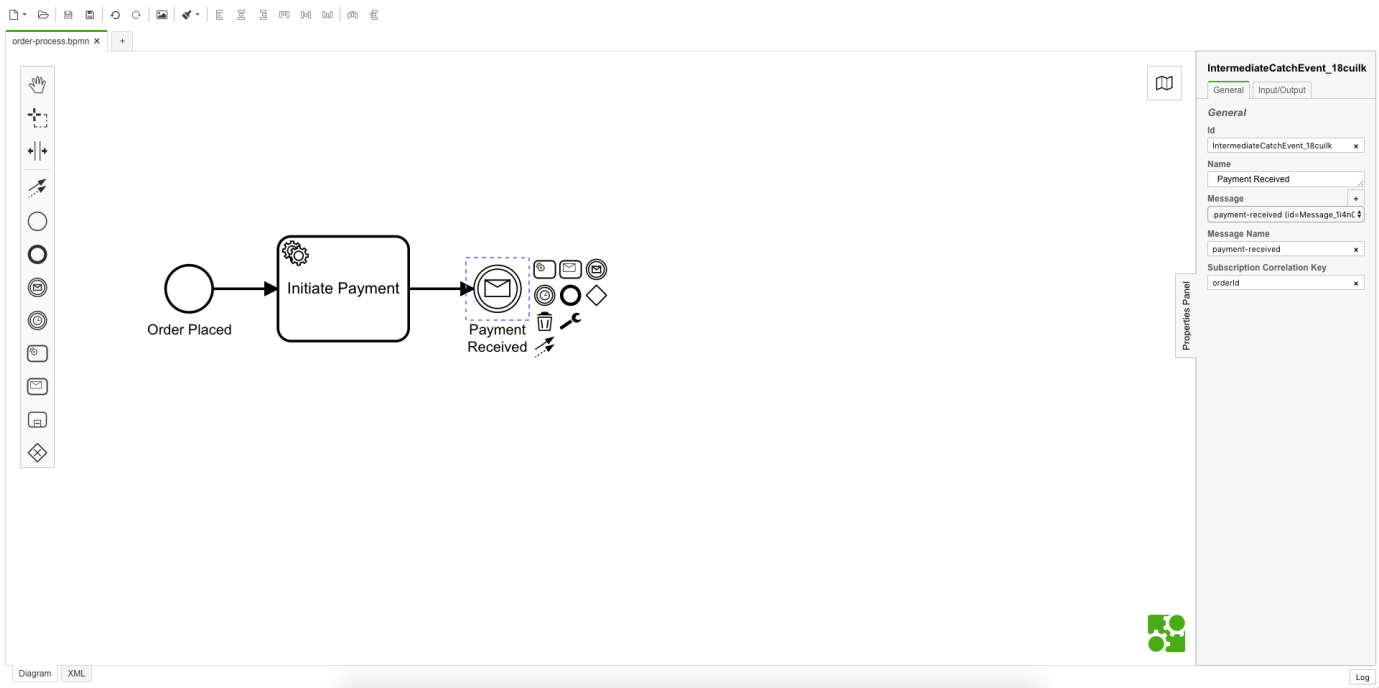
But how do we know which *specific workflow instance*--that is, which customer order--a message refers to? That's where Subscription Correlation Key comes in. The Subscription Correlation Key is a unique ID present in both the workflow instance payload and the message sent to Zeebe.

We'll use `orderId` for our correlation key.

Go ahead and add `orderId` to the Subscription Correlation Key field.

When we create a workflow instance, we need to be sure to include `orderId` as a variable, and we also need to provide `orderId` as a correlation key when we send a message.

Here's what you should see in the Modeler:

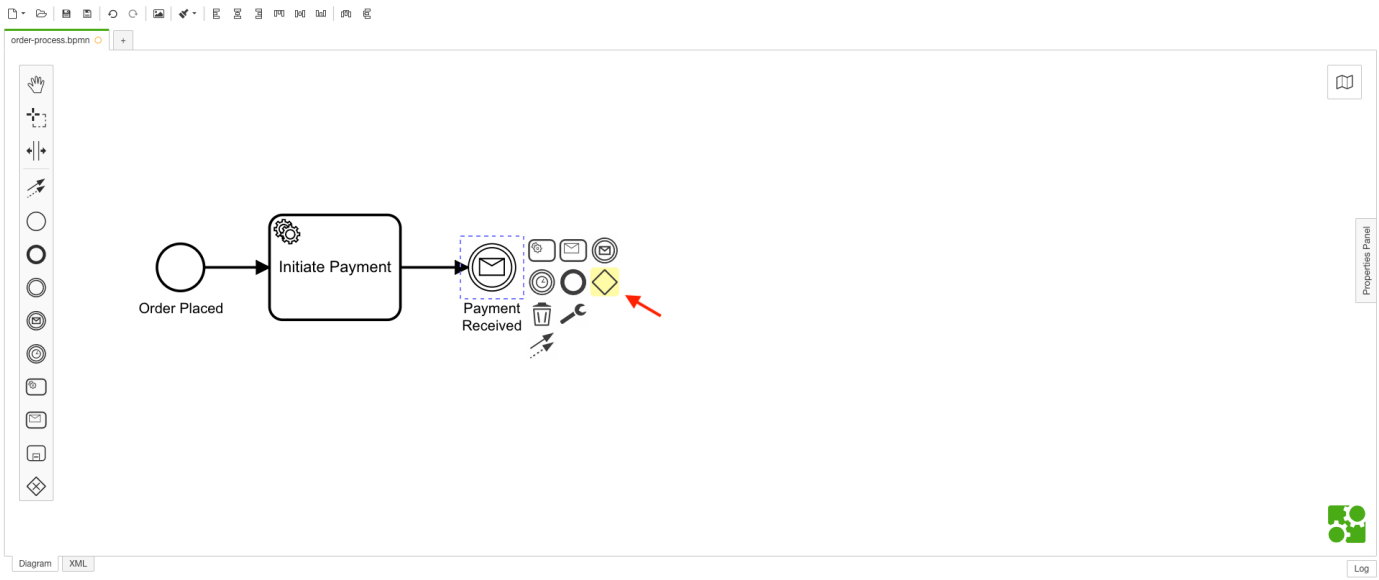


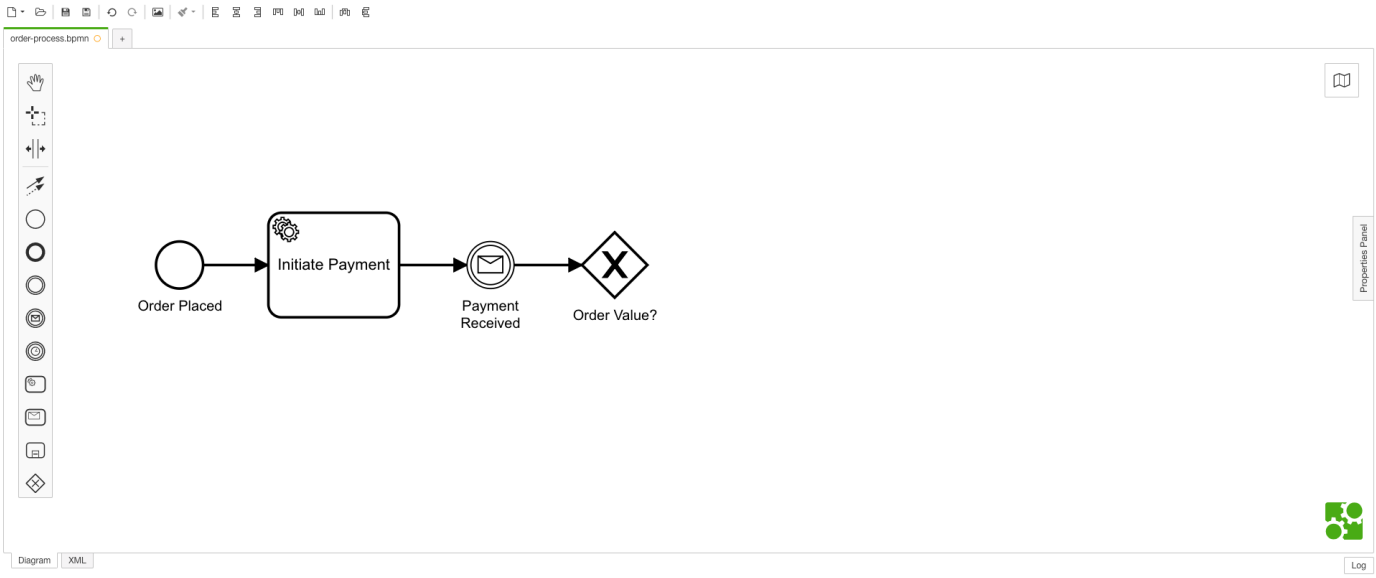
Next, we'll add an Exclusive (XOR) Gateway to our workflow model. The Exclusive Gateway is used to make a data-based decision about which path a workflow instance should follow. In this case, we want to ship items *with* insurance if total order value is greater than or equal to $100 and ship *without* insurance.

That means that when we create a workflow instance, we'll need to include order value as an instance variable. But we'll come to that later.

First, let's take the necessary steps to configure our workflow model to make this decision. To add the gateway:

- Click on the Message Event you just created
- Select the Gateway (diamond-shaped) symbol - the Exclusive Gateway is the default when you add a new gateway to a model
- Double-click on the gateway and add a label "Order Value?" so that it's clear what we're using as our decision criteria

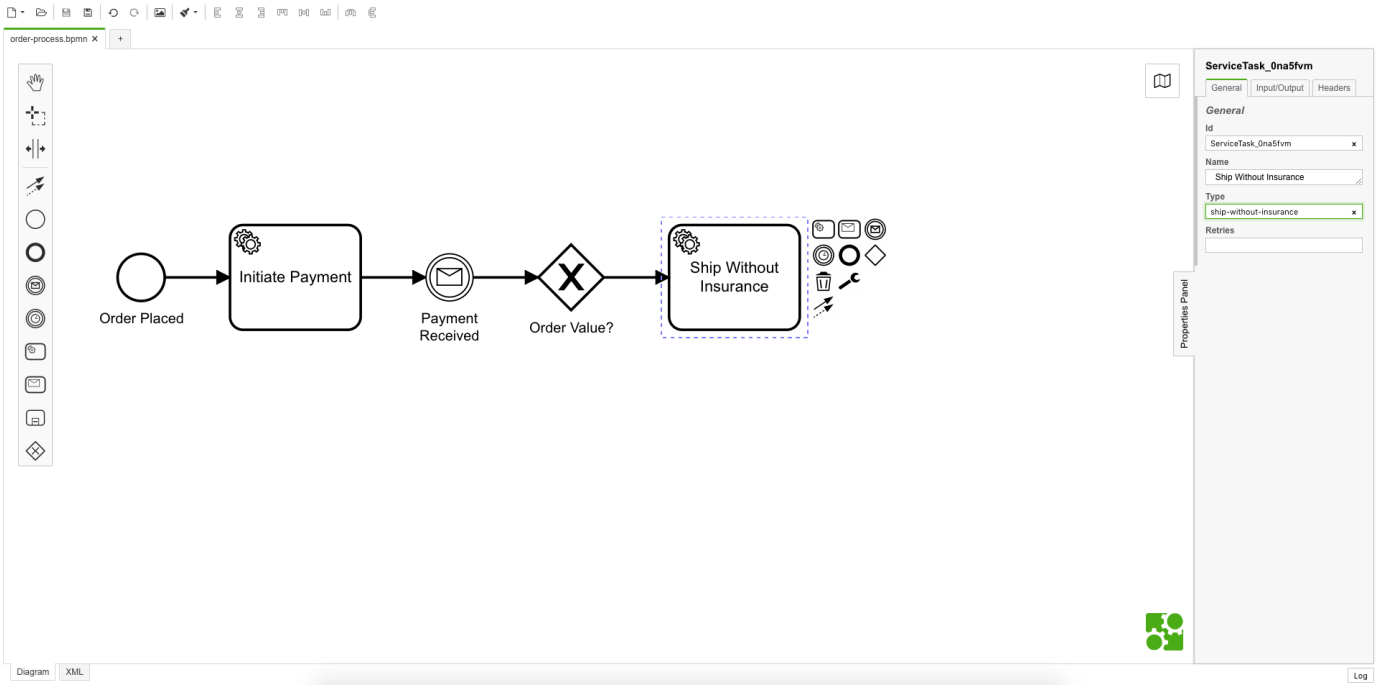




We'll add two outgoing Sequence Flows from this Exclusive Gateway that lead to two different Service Tasks. Each Sequence Flow will have a data-based condition that's evaluated in the context of the workflow instance payload.

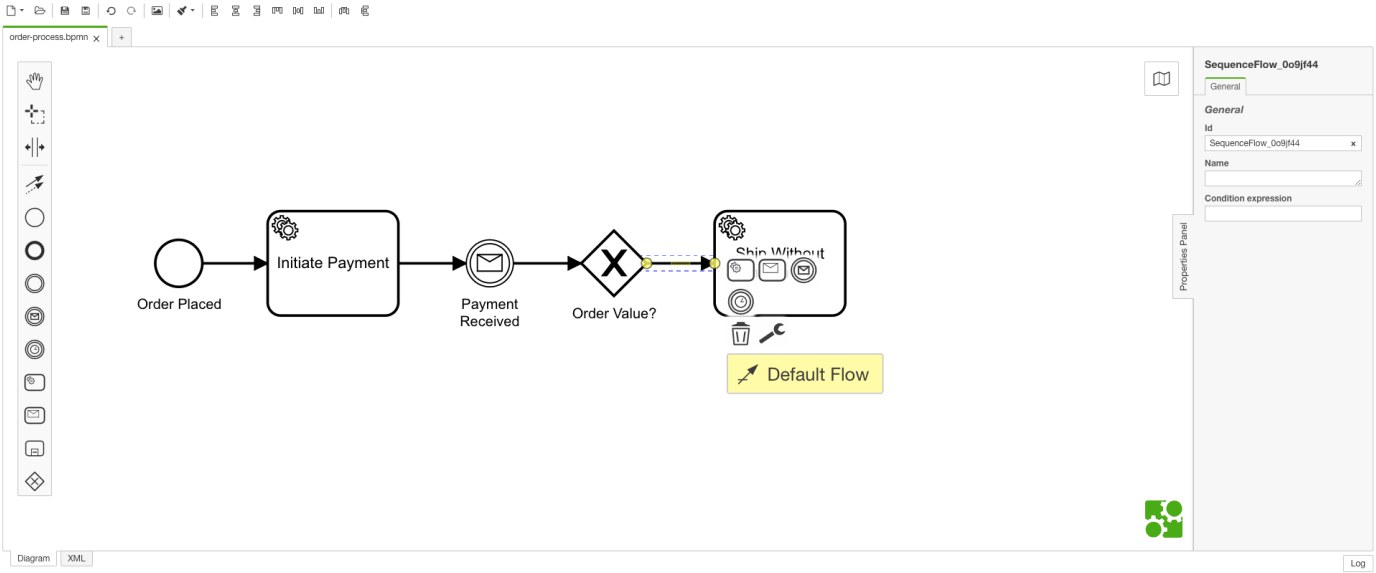
Next, we need to:

- Select the gateway and add a new Service Task to the model.
- Label the task "Ship Without Insurance"
- Set the Type to `ship-without-insurance`



Whenever we use an Exclusive Gateway, we want to be sure to set a default flow, which in this case will be shipping without insurance:

- Select the Sequence Flow you just created from the gateway to the "Ship Without Insurance" Service Task
- Click on the wrench icon
- Choose "Default Flow"



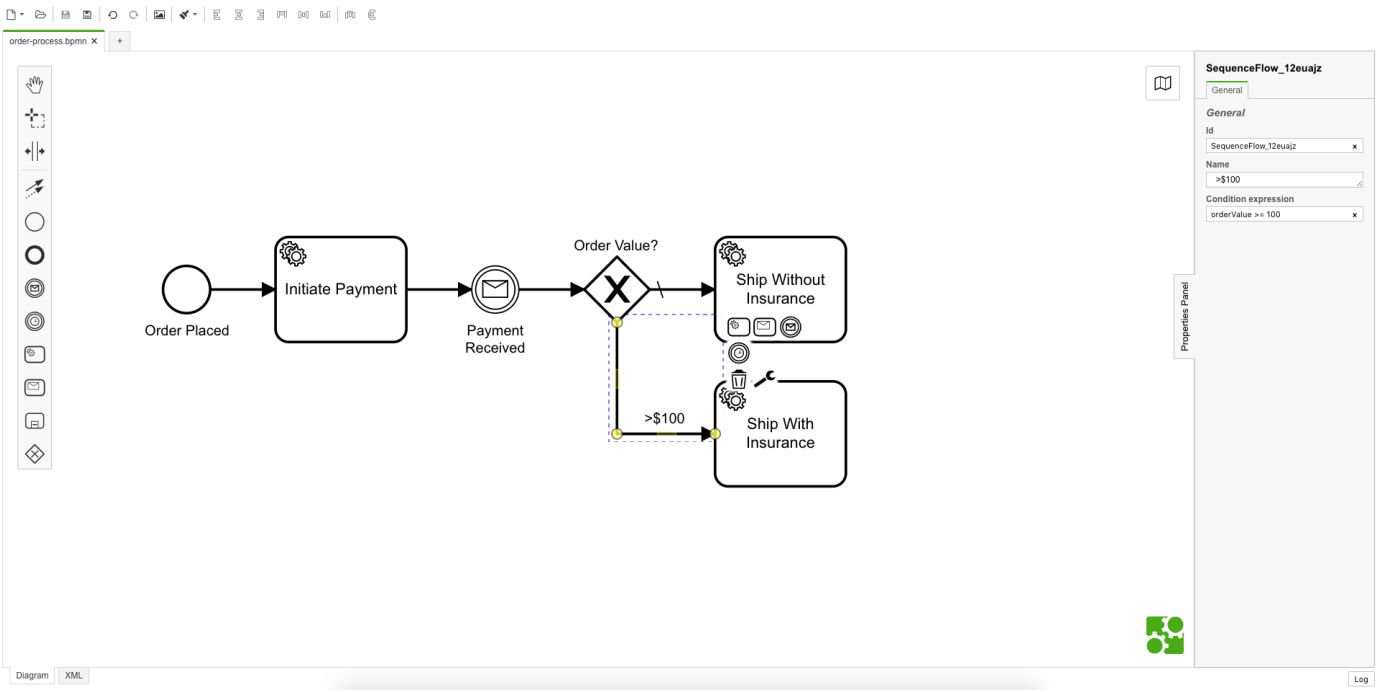
Now we're ready to add a *second* outgoing Sequence Flow and Service Task from the gateway:

- Select the gateway again

- Add another Service Task to the model
- Label it "Ship With Insurance"
- Set the type to `ship-with-insurance`

Next, we'll set a condition expression in the Sequence Flow leading to this "Ship With Insurance" Service Task:

- Click on the sequence flow and open the Properties panel
- Input `orderValue >= 100` in the "Condition expression" field in the Properties panel
- Double-click on the sequence flow to add a label "`>$100`"

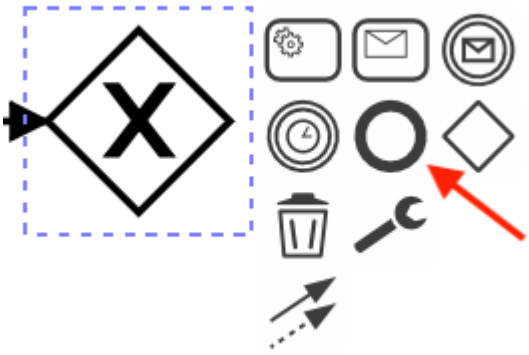


We're almost finished! To wrap things up, we'll:

- Select the "Ship Without Insurance" task
- Add another Exclusive Gateway to the model to merge the branches together again (a BPMN best practice in a model like this one).
- Select the "Ship With Insurance" task
- Add an outgoing sequence flow that connects to the second Exclusive Gateway you just created

The only BPMN element we need to add is an End Event:

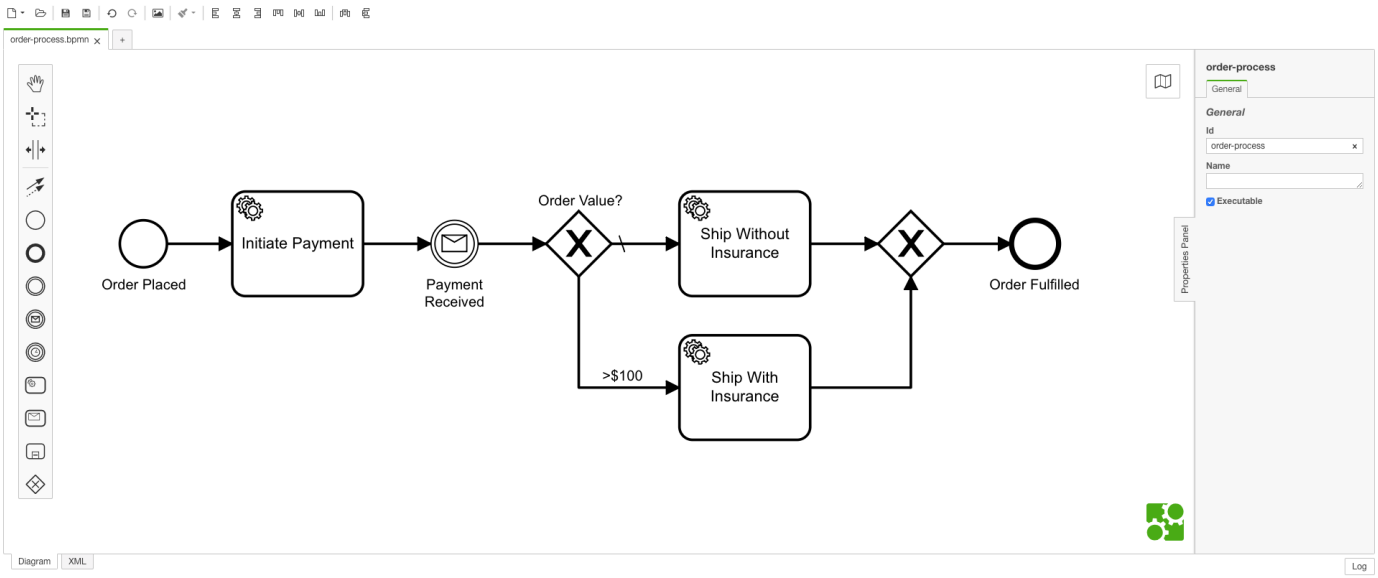
- Click on the second Exclusive Gateway
- Add an End Event
- Double-click on it to label it "Order Fulfilled"



Lastly, we'll change the process ID to something more descriptive than the default `Process_1` that you'll see in the Modeler:

- Haga clic en una parte en blanco del lienzo.
- Abra el panel de propiedades
- Cambiar el Id a `order-process`

Esto es lo que debería ver en el Modelador después de estas últimas actualizaciones:



Eso es todo por nuestro paso de modelado. Recuerde guardar el archivo una vez más para prepararse para implementar el flujo de trabajo en Zeebe, crear instancias de flujo de trabajo y completarlas.

Página siguiente: Implementar un flujo de trabajo >>

<< Página anterior: Configuración del tutorial