

Crear y completar instancias de flujo de trabajo

Vamos a crear 2 instancias de flujo de trabajo para este tutorial: una con un valor de pedido inferior a \$ 100 y otra con un valor de pedido superior o igual a \$ 100 para que podamos ver nuestra puerta de enlace XOR en acción.

Regrese a la ventana Terminal donde implementó el modelo de flujo de trabajo y ejecute el siguiente comando.

Nota: Los usuarios de Windows que desean ejecutar este comando usando cmd o Powershell tienen que escapar de las variables de manera diferente.

- cmd: `"{\\"orderId\\": 1234}"`
- Potencia Shell: `'{"\\"orderId\\"": 1234}'`

Linux

```
./bin/zbctl --insecure create instance order-process --variables '{"orderId": "1234", "orderValue":99}'
```

Mac

```
./bin/zbctl.darwin --insecure create instance order-process --variables '{"orderId": "1234", "orderValue":99}'
```

Windows (Powershell)

```
./bin/zbctl.exe --insecure create instance order-process --variables '{\\"orderId\\": \\"1234\\", \\"orderValue\\":99}'
```

Verás una respuesta como:

```
{
  "workflowKey": 1,
  "bpmnProcessId": "order-process",
  "version": 1,
  "workflowInstanceKey": 8
}
```

Esta primera instancia de flujo de trabajo que acabamos de crear representa un pedido de cliente único con orderId 1234 y orderValue 99 (o \$ 99).

En la misma ventana de Terminal, ejecute el comando:

Linux

```
./bin/zbctl --insecure create instance order-process --variables '{"orderId": "2345", "orderValue":100}'
```

Mac

```
./bin/zbctl.darwin --insecure create instance order-process --variables '{"orderId": "2345", "orderValue":100}'
```

Windows (Powershell)

```
./bin/zbctl.exe --insecure create instance order-process --variables '{"orderId\": \"2345\", \"orderValue\":100}'
```

Esta segunda instancia de flujo de trabajo que acabamos de crear representa un pedido de cliente único con orderId 2345 y orderValue 100 (o \$ 100).

Si vuelve a la interfaz de usuario Operate y actualiza la página, ahora debería ver dos instancias de flujo de trabajo (la insignia verde) esperando en la tarea Iniciar pago.

The screenshot shows the Zeebe Operate interface. On the left, the 'Filters' panel is set to 'order-process' and 'Version 1'. The main area displays the 'order-process' workflow diagram, which starts with 'Order Placed', followed by 'Initiate Payment', 'Payment Received', a decision 'Order Value?', and then two paths: 'Ship Without Insurance' and 'Ship With Insurance', both leading to 'Order Fulfilled'. Below the diagram, the 'Instances' table shows two running instances of 'order-process'.

Workflow	Instance Id	Version	Start Time	End Time	Actions
order-process	11	Version 1	4 Apr 2019 16:31:42	--	[Stop]
order-process	3	Version 1	4 Apr 2019 16:31:32	--	[Stop]

Tenga en cuenta que la instancia de flujo de trabajo no puede pasar esta primera tarea hasta que creamos un trabajador de trabajo para completar los `initiate-payment` trabajos. Entonces eso es exactamente lo que haremos a continuación.

Para volver a señalar esto: en un caso de uso de palabras reales, probablemente no crearía manualmente instancias de flujo de trabajo utilizando la CLI de Zeebe. Por el contrario, una instancia de flujo de trabajo se crearía mediante programación en respuesta a algún evento comercial, como un mensaje enviado a Zeebe después de que un cliente realiza un pedido. Y se podrían crear instancias a gran escala si, por ejemplo, muchos clientes realizaban

pedidos al mismo tiempo debido a una venta. Estamos usando la CLI aquí solo por simplicidad.

Actualmente tenemos dos instancias esperando en nuestra tarea "Iniciar pago", lo que significa que Zeebe ha creado dos trabajos con tipo `initiate-payment`.

zbctl proporciona un comando para generar trabajadores simples mediante un comando o script externo. El trabajador de trabajo recibirá la carga útil para cada trabajo como un objeto JSON en stdin y también debe devolver su resultado como objeto JSON en stdout si manejó el trabajo con éxito.

En este ejemplo, también usaremos el comando unix cat que solo genera lo que recibe en stdin.

Abra una nueva pestaña o ventana de Terminal, cambie al directorio de intermediario de Zeebe y use el siguiente comando para crear un trabajador de trabajo que trabaje en el `initiate-payment` trabajo.

Nota: Para usuarios de Windows, este comando no funciona con cmd ya que el `cat` comando no existe. Recomendamos utilizar Powershell o un shell bash-like para ejecutar este comando.

Linux

```
./bin/zbctl --insecure create worker initiate-payment --handler cat
```



Mac

```
./bin/zbctl.darwin --insecure create worker initiate-payment --handler cat
```



Ventanas

```
./bin/zbctl.exe --insecure create worker initiate-payment --handler "findstr .
```



Debería ver una respuesta en la línea de:

```
Activated job 12 with payload {"orderId":"2345","orderValue":100}  
Activated job 7 with payload {"orderId":"1234","orderValue":99}  
Handler completed job 12 with payload {"orderId":"2345","orderValue":100}  
Handler completed job 7 with payload {"orderId":"1234","orderValue":99}
```



Podemos ver que el trabajador activado activó y luego completó los dos `initiate-payment` trabajos disponibles. Puede cerrar el trabajador de trabajo si lo desea; no lo necesitará en el resto del tutorial.

Ahora vaya a la pestaña del navegador donde está ejecutando Operate. Debería ver que las instancias de flujo de trabajo han avanzado al evento Intermediate Message Catch y están esperando allí.

The screenshot shows the Zeebe UI interface. On the left, there's a 'Filters' sidebar with options for 'Running Instances' (Active, Incidents) and 'Finished Instances' (Completed, Canceled). The main area displays the 'order-process' workflow diagram, which starts with 'Order Placed', followed by 'Initiate Payment', then a message event 'Payment Received' (with a green circle and '2' indicating two instances). This leads to an XOR gateway 'Order Value?'. One path goes to 'Ship Without Insurance' and another to 'Ship With Insurance' (with a condition '>\$100'). Both paths merge at another XOR gateway before 'Order Fulfilled'. Below the diagram is a table of 'Instances'.

Workflow	Instance Id	Version	Start Time	End Time	Actions
order-process	11	Version 1	4 Apr 2019 16:31:42	--	[Stop]
order-process	3	Version 1	4 Apr 2019 16:31:32	--	[Stop]

Las instancias de flujo de trabajo esperarán en el Evento de captura de mensajes intermedios hasta que Zeebe reciba un mensaje y esté correlacionado con las instancias. Los mensajes se pueden publicar utilizando clientes de Zeebe, y también es posible que Zeebe se conecte a una cola de mensajes como Apache Kafka y correlacione los mensajes publicados allí con las instancias de flujo de trabajo.

zbctl también admite la publicación de mensajes, por lo que continuaremos usándolo en nuestra demostración. A continuación se muestra el comando que usaremos para publicar y correlacionar un mensaje. Verá que proporcionamos el mensaje "Nombre" que asignamos a este evento de mensaje en el Modelador de Zeebe, así como el Id. De pedido que incluimos en la carga útil de la instancia cuando la creamos.

Recuerde, `orderId` es la clave de correlación que configuramos en el Modelador al configurar el evento de mensaje. Zeebe requiere ambos campos para poder correlacionar un mensaje con una instancia de flujo de trabajo. Debido a que tenemos dos instancias de flujo de trabajo con dos distintas `orderId`, necesitaremos publicar dos mensajes. Ejecute estos dos comandos uno tras otro:

Linux

```
./bin/zbctl --insecure publish message "payment-received" --
correlationKey="1234"
./bin/zbctl --insecure publish message "payment-received" --
correlationKey="2345"
```

Mac

```
./bin/zbctl.darwin --insecure publish message "payment-received" --
correlationKey="1234"
./bin/zbctl.darwin --insecure publish message "payment-received" --
correlationKey="2345"
```

Ventanas

```
./bin/zbctl.exe --insecure publish message "payment-received" --
correlationKey="1234"
./bin/zbctl.exe --insecure publish message "payment-received" --
correlationKey="2345"
```



No verá una respuesta en su ventana de Terminal, pero si actualiza Operate, debería ver que los mensajes se correlacionaron correctamente y que una instancia de flujo de trabajo avanzó a la tarea "Enviar con seguro" y la otra avanzó a "Enviar sin seguro" tarea.

The screenshot shows the Zeebe Operate interface. On the left, there's a 'Filters' sidebar with 'order-process' selected. The main area displays the 'order-process' workflow diagram. The diagram starts with 'Order Placed', followed by 'Initiate Payment', then 'Payment Received'. A decision node 'Order Value?' branches into two paths: one for '≤ \$100' leading to 'Ship Without Insurance', and another for '> \$100' leading to 'Ship With Insurance'. Both paths converge at a final decision node before 'Order Fulfilled'. Below the diagram, a table lists running instances:

Workflow	Instance Id	Version	Start Time	End Time	Actions
order-process	11	Version 1	4 Apr 2019 16:31:42	--	[Stop]
order-process	3	Version 1	4 Apr 2019 16:31:32	--	[Stop]

La buena noticia es que esta visualización confirma que nuestra lógica de decisión funcionó como se esperaba: nuestra instancia de flujo de trabajo con `orderValue` \$ 100 se enviará con seguro, y nuestra instancia de flujo de trabajo con `orderValue` \$ 99 se enviará sin seguro.

Probablemente sepa lo que debe hacer a continuación. Continúe y abra una ventana de Terminal y cree un trabajador de trabajo para el `ship-without-insurance` tipo de trabajo.

Linux

```
./bin/zbctl --insecure create worker ship-without-insurance --handler cat
```



Mac

```
./bin/zbctl.darwin --insecure create worker ship-without-insurance --handler cat
```



Ventanas

```
./bin/zbctl.exe --insecure create worker ship-without-insurance --handler
"findstr ."
```



Debería ver una respuesta en la línea de:

```
Activated job 529 with payload {"orderId":"1234","orderValue":99}
Handler completed job 529 with payload {"orderId":"1234","orderValue":99}
```

Puede cerrar este trabajador ahora.

Seleccione la casilla de verificación "Instancias terminadas" en la parte inferior izquierda de Operar, actualice la página y ¡listo! Verá su primera instancia de flujo de trabajo Zeebe completada.

The screenshot shows the Zeebe Operator interface. On the left, the 'Filters' panel is expanded, showing 'Running Instances' and 'Finished Instances' sections. The 'Finished Instances' section has checkboxes for 'Completed' and 'Canceled'. The main area displays the 'order-process' workflow diagram, which includes steps like 'Order Placed', 'Initiate Payment', 'Payment Received', and decision points for 'Order Value?'. Below the diagram, the 'Instances' table lists two instances of the 'order-process' workflow.

Workflow	Instance Id	Version	Start Time	End Time	Actions
<input checked="" type="checkbox"/> order-process	11	Version 1	4 Apr 2019 16:31:42	--	
<input checked="" type="checkbox"/> order-process	3	Version 1	4 Apr 2019 16:31:32	4 Apr 2019 16:36:42	

Debido a que la tarea "Enviar con seguro" tiene un tipo de trabajo diferente, necesitamos crear un segundo trabajador que pueda asumir este trabajo.

Linux

```
./bin/zbctl --insecure create worker ship-with-insurance --handler cat
```

Mac

```
./bin/zbctl.darwin --insecure create worker ship-with-insurance --handler cat
```

Ventanas

```
./bin/zbctl.exe --insecure create worker ship-with-insurance --handler "findst
.*"
```

Debería ver una respuesta en la línea de:

```
Activated job 535 with payload {"orderId":"2345","orderValue":100}
Handler completed job 535 with payload {"orderId":"2345","orderValue":100}
```

También puedes cerrar a este trabajador.

Echemos un vistazo más en Operate para confirmar que ambas instancias de flujo de trabajo se han completado.

The screenshot displays the Zeebe Operate interface. On the left, the 'Filters' panel shows filters for 'order-process' and 'Version 1'. Below these are input fields for 'Instance Id(s)', 'Error Message', 'Start Date', and 'End Date'. A 'Flow Node' dropdown is also present. Checkboxes for 'Running Instances' (Active, Incidents) and 'Finished Instances' (Completed, Canceled) are visible, along with a 'Reset Filters' button.

The main area shows the 'order-process' workflow diagram. It starts with 'Order Placed', followed by 'Initiate Payment', then 'Payment Received'. A decision node 'Order Value?' branches into 'Ship Without Insurance' (if >\$100) and 'Ship With Insurance'. Both paths lead to a final 'Order Completed' state.

Below the diagram is the 'Instances' table:

Workflow	Instance Id	Version	Start Time	End Time	Actions
order-process	11	Version 1	4 Apr 2019 16:31:42	4 Apr 2019 16:38:01	
order-process	3	Version 1	4 Apr 2019 16:31:32	4 Apr 2019 16:36:42	

At the bottom, there is a 'Create Selection' button.

¡Hurra! ¡Has completado el tutorial! Felicidades.

En la sección siguiente y final, le indicaremos los recursos que creemos que le resultarán útiles a medida que continúe trabajando con Zeebe.

Página siguiente: Próximos pasos y recursos >>

<< Página anterior: Implementar un flujo de trabajo