FLOWX.AI

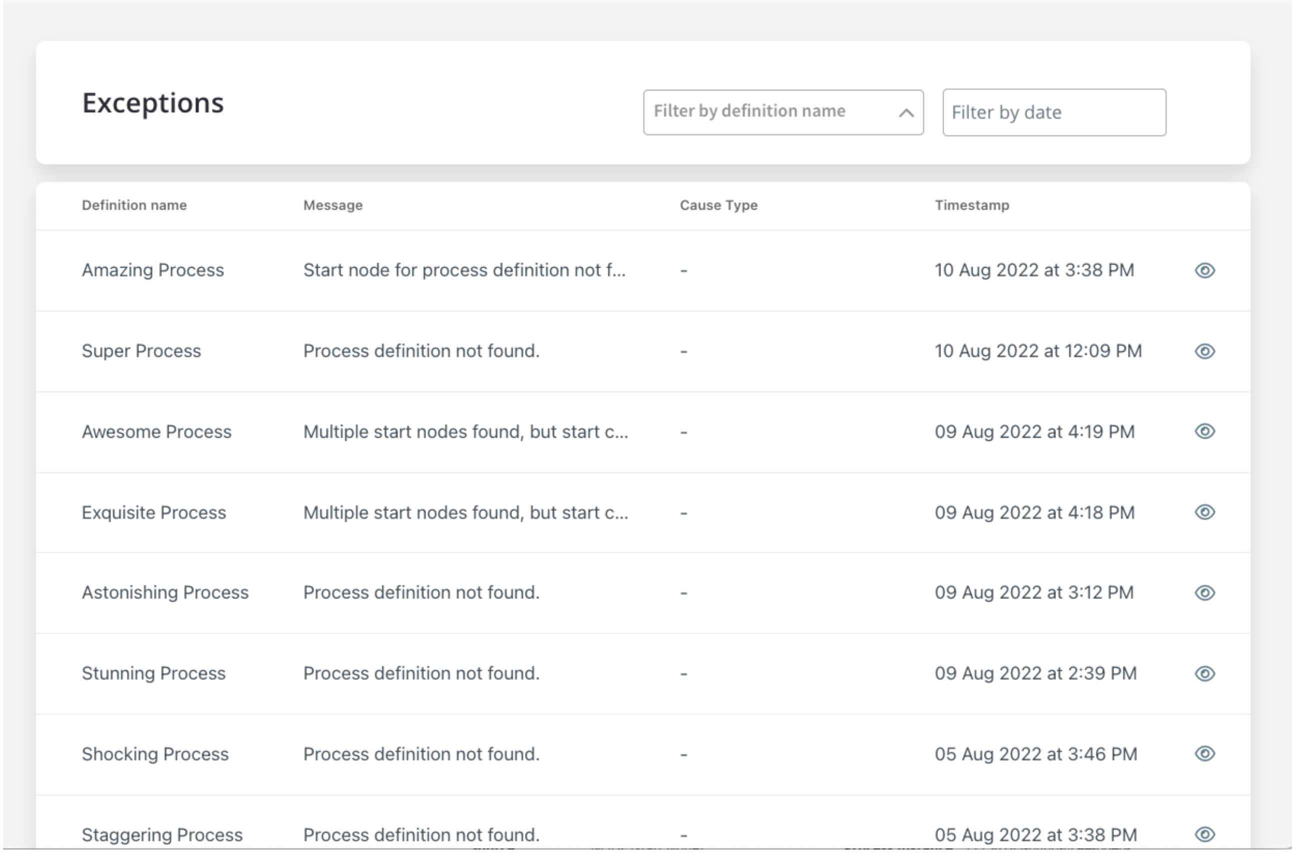
BUILDING BLOCKS / Process Designer / Active process / failed-process-start

Contents

BUILDING BLOCKS / Process Designer / Active process / Failed process start

Exceptions

Exceptions are types of errors meant to help you debug a failure in the execution of a process.

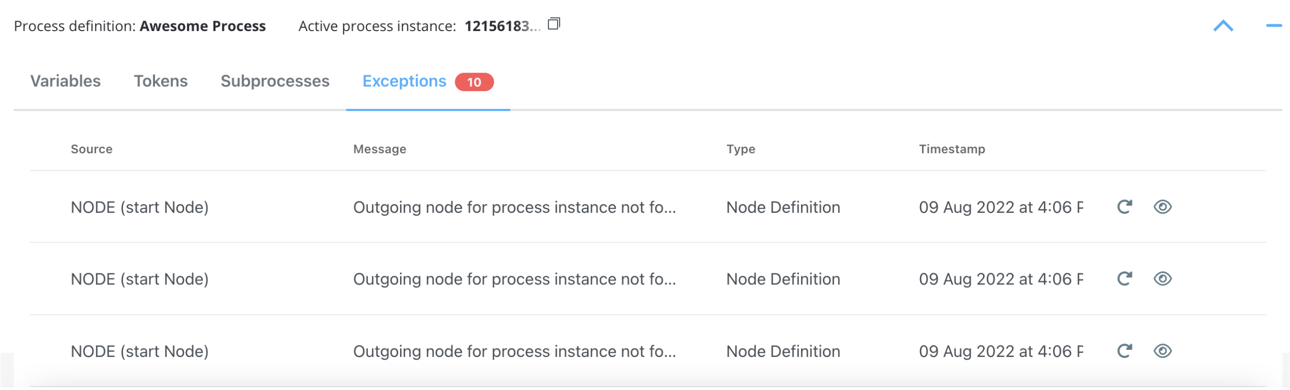


Exceptions

Definition name	Message	Cause Type	Timestamp
Amazing Process	Start node for process definition not f...	-	10 Aug 2022 at 3:38 PM
Super Process	Process definition not found.	-	10 Aug 2022 at 12:09 PM
Awesome Process	Multiple start nodes found, but start c...	-	09 Aug 2022 at 4:19 PM
Exquisite Process	Multiple start nodes found, but start c...	-	09 Aug 2022 at 4:18 PM
Astonishing Process	Process definition not found.	-	09 Aug 2022 at 3:12 PM
Stunning Process	Process definition not found.	-	09 Aug 2022 at 2:39 PM
Shocking Process	Process definition not found.	-	05 Aug 2022 at 3:46 PM
Staggering Process	Process definition not found.	-	05 Aug 2022 at 3:38 PM

Exceptions can be accessed from multiple places:

- **Failed process start** tab from **Active process** menu in FLOWX Designer
- **Process Status** view, accessible from **Process instances** list in FLOWX Designer



Process definition: **Awesome Process** Active process instance: **12156183...**

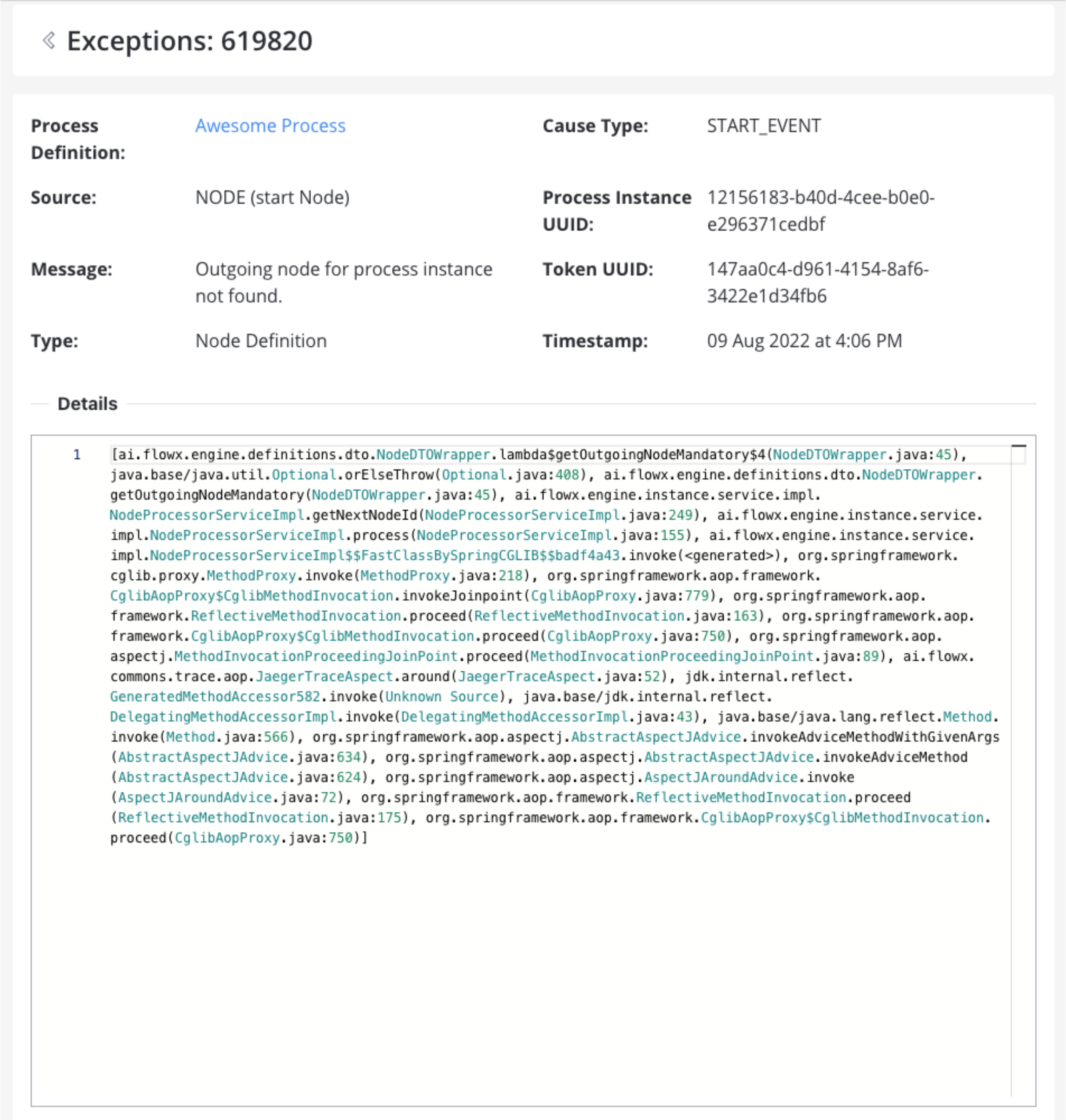
Source	Message	Type	Timestamp
NODE (start Node)	Outgoing node for process instance not fo...	Node Definition	09 Aug 2022 at 4:06 F
NODE (start Node)	Outgoing node for process instance not fo...	Node Definition	09 Aug 2022 at 4:06 F
NODE (start Node)	Outgoing node for process instance not fo...	Node Definition	09 Aug 2022 at 4:06 F

⚠ CAUTION

If you open a process instance and it does not contain exceptions, the **Exceptions** tab will not be displayed.

Exceptions data

When you click **view** button, a detailed exception will be displayed.



Exceptions: 619820

Process Definition:	Awesome Process	**Cause Type:**	START_EVENT
Source:	NODE (start Node)	**Process Instance UUID:**	12156183-b40d-4cee-b0e0-e296371cedbf
Message:	Outgoing node for process instance not found.	**Token UUID:**	147aa0c4-d961-4154-8af6-3422e1d34fb6
Type:	Node Definition	**Timestamp:**	09 Aug 2022 at 4:06 PM

Details

```
[ai.flowx.engine.definitions.dto.NodeDTOWrapper.lambda$getOutgoingNodeMandatory$4(NodeDTOWrapper.java:45),
java.base/java.util.Optional.orElseThrow(Optional.java:408), ai.flowx.engine.definitions.dto.NodeDTOWrapper.
getOutgoingNodeMandatory(NodeDTOWrapper.java:45), ai.flowx.engine.instance.service.impl.
NodeProcessorServiceImpl.getNextNodeId(NodeProcessorServiceImpl.java:249), ai.flowx.engine.instance.service.
impl.NodeProcessorServiceImpl.process(NodeProcessorServiceImpl.java:155), ai.flowx.engine.instance.service.
impl.NodeProcessorServiceImpl$$FastClassBySpringCGLIB$$badf4a43.invoke(<generated>), org.springframework.
cglib.proxy.MethodProxy.invoke(MethodProxy.java:218), org.springframework.aop.framework.
CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:779), org.springframework.aop.
framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163), org.springframework.aop.
framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:750), org.springframework.aop.
aspectj.MethodInvocationProceedingJoinPoint.proceed(MethodInvocationProceedingJoinPoint.java:89), ai.flowx.
commons.trace.aop.JaegerTraceAspect.around(JaegerTraceAspect.java:52), jdk.internal.reflect.
GeneratedMethodAccessor582.invoke(Unknown Source), java.base/jdk.internal.reflect.
DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43), java.base/java.lang.reflect.Method.
invoke(Method.java:566), org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethodWithGivenArgs
(AbstractAspectJAdvice.java:634), org.springframework.aop.aspectj.AbstractAspectJAdvice.invokeAdviceMethod
(AbstractAspectJAdvice.java:624), org.springframework.aop.aspectj.AspectJAroundAdvice.invoke
(AspectJAroundAdvice.java:72), org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:175), org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.
proceed(CglibAopProxy.java:750)]
```

- **Process Definition** - the process where the exception was thrown
- **Source** - the source of the exception (see the possible type of sources below)
- **Message** - a hint type of message to help you understand what's wrong with your process

- **Type** - exception type
- **Cause Type** - cause type (or the name of the node)
- **Process Instance UUID** - process instance unique identifier
- **Token UUID** - token unique identifier
- **Timestamp** - default format: `yyyy-MM-dd'T'HH:mm:ss.SSSZ`
- **Details** - stack trace (a **stack trace** is a list of the method calls that the process was in the middle of when an **Exception** was thrown)

Possible sources:

- Action
- Node
- Subprocess
- Process Definition

Exceptions type

Based on the exception type, there are multiple causes that could make a process fail. Here are some examples:

Type	Cause
Business Rule Evaluation	when executing action rules fails for any reason
Condition Evaluation	when executing action conditions

Type	Cause
Engine	when the connection with the database fails when the connection with Redis fails
Definition	misconfigurations: process def name, subprocess parent process id value, start node condition missing
Node	when an outgoing node can't be found (missing sequence etc)
Gateway Evaluation	when the token can't pass a gateway for any reason, possible causes: • missing sequence/node • failed node rule
Subprocess	exceptions will be saved for them just like for any other process, parent process ID will also be saved (we can use this to link them when displaying exceptions)

Was this page helpful?