A `ChannelSink` is an internal component that is supposed to be implemented by a transport provider. (Most users will not see this type in their code.)

The **exceptionCaught** method is implemented in the `AbstractChannelSink` and is invoked by `ChannelPipeline` when an exception is raised by a `ChannelHandler` that is processing a `ChannelEvent`.

The **eventSunk** method in `ChannelSink` is invoked by `ChannelPipeline` when a downstream `ChannelEvent` has reached its terminal (the head of the pipeline).

CHANNEL SINK

ChannelSink
+eventSunk(pipeline: ChannelPipeline, e: ChannelEvent) +exceptionCaught(pipeline: ChannelPipeline, e: ChannelEvent)

DiscardingChannelSink —○ **DefaultChannelPipeline**

EmbeddedChannelSink —○ **AbstractCodecEmbedder**

<<interface>>
ChannelSink

AbstractChannelSink

- HttpTunnelingClientSocketPipelineSink
- LocalClientChannelSink
- LocalServerChannelSink
- NioClientSocketPipelineSink
- NioDatagramPipelineSink
- NioServerSocketPipelineSink
- OioClientSocketPipelineSink
- OioDatagramPipelineSink
- OioServerSocketPipelineSink
- OioServerSocketPipelineSink
- XnioClientChannelSink
- XnioServerChannelSink

I/O Request via Channel or ChannelHandlerContext

LAST	Upstream Handler	Downstream Handler	LAST
	Upstream Handler	Downstream Handler	
	Going Upstream	Going Downstream	
	Upstream Handler	Downstream Handler	
FIRST	Upstream Handler	Downstream Handler	FIRST

I/O Threads (Transport Implementation)

NODE 1

NODE 2