

**Module** java.base  
**Package** java.util.stream

## Interface Gatherer.Integrator<A,T,R>

**Type Parameters:**  
A - the type of state used by this integrator  
T - the type of elements this integrator consumes  
R - the type of results this integrator can produce

**All Known Subinterfaces:**  
`Gatherer.Integrator.GreedyPREVIEW<A,T,R>`

**Enclosing interface:**  
`GathererPREVIEW<T,A,R>`

**Functional Interface:**  
This is a functional interface and can therefore be used as the assignment target for a lambda expression or method reference.

@FunctionalInterface  
public static interface **Gatherer.Integrator**<A,T,R>

**Integrator is a preview API of the Java platform.**  
*Programs can only use Integrator when preview features are enabled.*  
*Preview features may be removed in a future release, or upgraded to permanent features of the Java platform.*

An Integrator receives elements and processes them, optionally using the supplied state, and optionally sends incremental results downstream.

**Since:**  
22

### Nested Class Summary

Nested Classes		
Modifier and Type	Interface	Description
static interface	<b>Gatherer.Integrator.Greedy<sup>PREVIEW</sup>&lt;A,T,R&gt;</b>	<b>Preview.</b> Greedy Integrators consume all their input, and may only relay that the downstream does not want more elements.

### Method Summary

All Methods	Static Methods	Instance Methods	Abstract Methods
Modifier and Type	Method		Description
boolean	<b>integrate</b> (A state, T element, <b>Gatherer.Downstream<sup>PREVIEW</sup>&lt;? super R&gt; downstream</b> )		Performs an action given: the current state, the next element, and a downstream object; potentially inspecting and/or updating the state, optionally sending any number of elements downstream -- and then returns whether more elements are to be consumed or not.
static <A, T, R>	<b>Gatherer.Integrator<sup>PREVIEW</sup>&lt;A,T,R&gt;</b>	<b>of</b> ( <b>Gatherer.Integrator<sup>PREVIEW</sup>&lt;A,T,R&gt; integrator</b> )	Factory method for turning Integrator-shaped lambdas into Integrators.
static <A, T, R>	<b>Gatherer.Integrator.Greedy<sup>PREVIEW</sup>&lt;A,T,R&gt;</b>	<b>ofGreedy</b> ( <b>Gatherer.Integrator.Greedy<sup>PREVIEW</sup>&lt;A,T,R&gt; greedy</b> )	Factory method for turning Integrator-shaped lambdas into <b>Gatherer.Integrator.Greedy<sup>PREVIEW</sup></b> Integrators.

### Method Details

integrate
<pre>boolean integrate(A state,                   T element,                   Gatherer.Downstream<sup>PREVIEW</sup>&lt;? super R&gt; downstream)</pre> <p>Performs an action given: the current state, the next element, and a downstream object; potentially inspecting and/or updating the state, optionally sending any number of elements downstream -- and then returns whether more elements are to be consumed or not.</p> <p><b>Parameters:</b></p> <p>state - The state to integrate into</p> <p>element - The element to integrate</p> <p>downstream - The downstream object of this integration</p> <p><b>Returns:</b></p> <p>true if subsequent integration is desired, false if not</p>
of
<pre>static &lt;A, T, R&gt; Gatherer.Integrator<sup>PREVIEW</sup>&lt;A,T,R&gt; of(Gatherer.Integrator<sup>PREVIEW</sup>&lt;A,T,R&gt; integrator)</pre> <p>Factory method for turning Integrator-shaped lambdas into Integrators.</p> <p><b>Type Parameters:</b></p> <p>A - the type of state used by this integrator</p> <p>T - the type of elements this integrator receives</p> <p>R - the type of results this integrator can produce</p> <p><b>Parameters:</b></p> <p>integrator - a lambda to create as Integrator</p> <p><b>Returns:</b></p> <p>the given lambda as an Integrator</p>
ofGreedy
<pre>static &lt;A, T, R&gt; Gatherer.Integrator.Greedy<sup>PREVIEW</sup>&lt;A,T,R&gt; ofGreedy(Gatherer.Integrator.Greedy<sup>PREVIEW</sup>&lt;A,T,R&gt; greedy)</pre> <p>Factory method for turning Integrator-shaped lambdas into <b>Gatherer.Integrator.Greedy<sup>PREVIEW</sup></b> Integrators.</p> <p><b>Type Parameters:</b></p> <p>A - the type of state used by this integrator</p> <p>T - the type of elements this integrator receives</p> <p>R - the type of results this integrator can produce</p> <p><b>Parameters:</b></p> <p>greedy - a lambda to create as Integrator:Greedy</p> <p><b>Returns:</b></p> <p>the given lambda as a Greedy Integrator</p>