Flutter resumen documentación

# Async widgets

## future builder

A widget that builds itself based on the latest snapshot of interaction with a [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html).

### Managing the future

The [future](https://api.flutter.dev/flutter/widgets/FutureBuilder/future.html) must have been obtained earlier, e.g. during [State.initState](https://api.flutter.dev/flutter/widgets/State/initState.html), [State.didUpdateWidget](https://api.flutter.dev/flutter/widgets/State/didUpdateWidget.html), or [State.didChangeDependencies](https://api.flutter.dev/flutter/widgets/State/didChangeDependencies.html). It must not be created during the [State.build](https://api.flutter.dev/flutter/widgets/State/build.html) or [StatelessWidget.build](https://api.flutter.dev/flutter/widgets/StatelessWidget/build.html) method call when constructing the [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html). If the [future](https://api.flutter.dev/flutter/widgets/FutureBuilder/future.html) is created at the same time as the [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html), then every time the [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html)'s parent is rebuilt, the asynchronous task will be restarted.

A general guideline is to assume that every build method could get called every frame, and to treat omitted calls as an optimization.

### Timing

Widget rebuilding is scheduled by the completion of the future, using [State.setState](https://api.flutter.dev/flutter/widgets/State/setState.html), but is otherwise decoupled from the timing of the future. The [builder](https://api.flutter.dev/flutter/widgets/FutureBuilder/builder.html) callback is called at the discretion of the Flutter pipeline, and will thus receive a timing-dependent sub-sequence of the snapshots that represent the interaction with the future.

A side-effect of this is that providing a new but already-completed future to a [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html) will result in a single frame in the [ConnectionState.waiting](https://api.flutter.dev/flutter/widgets/ConnectionState.html) state. This is because there is no way to synchronously determine that a [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html) has already completed.

### Builder contract

For a future that completes successfully with data, assuming [initialData](https://api.flutter.dev/flutter/widgets/FutureBuilder/initialData.html) is null, the [builder](https://api.flutter.dev/flutter/widgets/FutureBuilder/builder.html) will be called with either both or only the latter of the following snapshots:

* AsyncSnapshot<String>.withData(ConnectionState.waiting, null)
* AsyncSnapshot<String>.withData(ConnectionState.done, 'some data')

If that same future instead completed with an error, the [builder](https://api.flutter.dev/flutter/widgets/FutureBuilder/builder.html) would be called with either both or only the latter of:

* AsyncSnapshot<String>.withData(ConnectionState.waiting, null)
* AsyncSnapshot<String>.withError(ConnectionState.done, 'some error', someStackTrace)

The initial snapshot data can be controlled by specifying [initialData](https://api.flutter.dev/flutter/widgets/FutureBuilder/initialData.html). You would use this facility to ensure that if the [builder](https://api.flutter.dev/flutter/widgets/FutureBuilder/builder.html) is invoked before the future completes, the snapshot carries data of your choice rather than the default null value.

The data and error fields of the snapshot change only as the connection state field transitions from waiting to done, and they will be retained when changing the [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html) configuration to another future. If the old future has already completed successfully with data as above, changing configuration to a new future results in snapshot pairs of the form:

* AsyncSnapshot<String>.withData(ConnectionState.none, 'data of first future')
* AsyncSnapshot<String>.withData(ConnectionState.waiting, 'data of second future')

In general, the latter will be produced only when the new future is non-null, and the former only when the old future is non-null.

A [FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html) behaves identically to a [StreamBuilder](https://api.flutter.dev/flutter/widgets/StreamBuilder-class.html) configured with future?.asStream(), except that snapshots with ConnectionState.active may appear for the latter, depending on how the stream is implemented.

### Constructors

[FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder/FutureBuilder.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<T>? future, T? initialData, required [AsyncWidgetBuilder](https://api.flutter.dev/flutter/widgets/AsyncWidgetBuilder.html)<T> builder})

Creates a widget that builds itself based on the latest snapshot of interaction with a [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html).

const

### Properties

[builder](https://api.flutter.dev/flutter/widgets/FutureBuilder/builder.html) → [AsyncWidgetBuilder](https://api.flutter.dev/flutter/widgets/AsyncWidgetBuilder.html)<T>

The build strategy currently used by this builder.

final

[future](https://api.flutter.dev/flutter/widgets/FutureBuilder/future.html) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<T>?

The asynchronous computation to which this builder is currently connected, possibly null.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[initialData](https://api.flutter.dev/flutter/widgets/FutureBuilder/initialData.html) → T?

The data that will be used to create the snapshots provided until a non-null [future](https://api.flutter.dev/flutter/widgets/FutureBuilder/future.html) has completed.

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

### Methods

[createElement](https://api.flutter.dev/flutter/widgets/StatefulWidget/createElement.html)() → [StatefulElement](https://api.flutter.dev/flutter/widgets/StatefulElement-class.html)

Creates a [StatefulElement](https://api.flutter.dev/flutter/widgets/StatefulElement-class.html) to manage this widget's location in the tree.

inherited

[createState](https://api.flutter.dev/flutter/widgets/FutureBuilder/createState.html)() → [State](https://api.flutter.dev/flutter/widgets/State-class.html)<[FutureBuilder](https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html)<T>>

Creates the mutable state for this widget at a given location in the tree.

override

[debugDescribeChildren](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/debugDescribeChildren.html)() → [List](https://api.flutter.dev/flutter/dart-core/List-class.html)<[DiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticsNode-class.html)>

Returns a list of DiagnosticsNode objects describing this node's children.

inherited

[debugFillProperties](https://api.flutter.dev/flutter/widgets/Widget/debugFillProperties.html)([DiagnosticPropertiesBuilder](https://api.flutter.dev/flutter/foundation/DiagnosticPropertiesBuilder-class.html) properties) → void

Add additional properties associated with the node.

inherited

[noSuchMethod](https://api.flutter.dev/flutter/dart-core/Object/noSuchMethod.html)([Invocation](https://api.flutter.dev/flutter/dart-core/Invocation-class.html) invocation) → dynamic

Invoked when a nonexistent method or property is accessed.

inherited

[toDiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toDiagnosticsNode.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html)? name, [DiagnosticsTreeStyle](https://api.flutter.dev/flutter/foundation/DiagnosticsTreeStyle.html)? style}) → [DiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticsNode-class.html)

Returns a debug representation of the object that is used by debugging tools and by [DiagnosticsNode.toStringDeep](https://api.flutter.dev/flutter/foundation/DiagnosticsNode/toStringDeep.html).

inherited

[toString](https://api.flutter.dev/flutter/foundation/Diagnosticable/toString.html)({[DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.info}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

A string representation of this object.

inherited

[toStringDeep](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toStringDeep.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html) prefixLineOne = '', [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? prefixOtherLines, [DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.debug}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

Returns a string representation of this node and its descendants.

inherited

[toStringShallow](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toStringShallow.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html) joiner = ', ', [DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.debug}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

Returns a one-line detailed description of the object.

inherited

[toStringShort](https://api.flutter.dev/flutter/widgets/Widget/toStringShort.html)() → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

A short, textual description of this widget.

inherited

**Operators**

[operator ==](https://api.flutter.dev/flutter/widgets/Widget/operator_equals.html)([Object](https://api.flutter.dev/flutter/dart-core/Object-class.html) other) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

The equality operator.

inherited

**Static Properties**

[debugRethrowError](https://api.flutter.dev/flutter/widgets/FutureBuilder/debugRethrowError.html) ↔ [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the latest error received by the asynchronous computation should be rethrown or swallowed. This property is useful for debugging purposes.

read / write

## stream builder

Widget that builds itself based on the latest snapshot of interaction with a [Stream](https://api.flutter.dev/flutter/dart-async/Stream-class.html).

Widget rebuilding is scheduled by each interaction, using [State.setState](https://api.flutter.dev/flutter/widgets/State/setState.html), but is otherwise decoupled from the timing of the stream. The [builder](https://api.flutter.dev/flutter/widgets/StreamBuilder/builder.html) is called at the discretion of the Flutter pipeline, and will thus receive a timing-dependent sub-sequence of the snapshots that represent the interaction with the stream.

As an example, when interacting with a stream producing the integers 0 through 9, the [builder](https://api.flutter.dev/flutter/widgets/StreamBuilder/builder.html) may be called with any ordered sub-sequence of the following snapshots that includes the last one (the one with ConnectionState.done):

* AsyncSnapshot<int>.withData(ConnectionState.waiting, null)
* AsyncSnapshot<int>.withData(ConnectionState.active, 0)
* AsyncSnapshot<int>.withData(ConnectionState.active, 1)
* ...
* AsyncSnapshot<int>.withData(ConnectionState.active, 9)
* AsyncSnapshot<int>.withData(ConnectionState.done, 9)

The actual sequence of invocations of the [builder](https://api.flutter.dev/flutter/widgets/StreamBuilder/builder.html) depends on the relative timing of events produced by the stream and the build rate of the Flutter pipeline.

Changing the [StreamBuilder](https://api.flutter.dev/flutter/widgets/StreamBuilder-class.html) configuration to another stream during event generation introduces snapshot pairs of the form:

* AsyncSnapshot<int>.withData(ConnectionState.none, 5)
* AsyncSnapshot<int>.withData(ConnectionState.waiting, 5)

The latter will be produced only when the new stream is non-null, and the former only when the old stream is non-null.

The stream may produce errors, resulting in snapshots of the form:

* AsyncSnapshot<int>.withError(ConnectionState.active, 'some error', someStackTrace)

The data and error fields of snapshots produced are only changed when the state is ConnectionState.active.

The initial snapshot data can be controlled by specifying [initialData](https://api.flutter.dev/flutter/widgets/StreamBuilder/initialData.html). This should be used to ensure that the first frame has the expected value, as the builder will always be called before the stream listener has a chance to be processed.

### Constructors

[StreamBuilder](https://api.flutter.dev/flutter/widgets/StreamBuilder/StreamBuilder.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, T? initialData, required [Stream](https://api.flutter.dev/flutter/dart-async/Stream-class.html)<T>? stream, required [AsyncWidgetBuilder](https://api.flutter.dev/flutter/widgets/AsyncWidgetBuilder.html)<T> builder})

Creates a new [StreamBuilder](https://api.flutter.dev/flutter/widgets/StreamBuilder-class.html) that builds itself based on the latest snapshot of interaction with the specified [stream](https://api.flutter.dev/flutter/widgets/StreamBuilderBase/stream.html) and whose build strategy is given by [builder](https://api.flutter.dev/flutter/widgets/StreamBuilder/builder.html).

const

### Properties

[builder](https://api.flutter.dev/flutter/widgets/StreamBuilder/builder.html) → [AsyncWidgetBuilder](https://api.flutter.dev/flutter/widgets/AsyncWidgetBuilder.html)<T>

The build strategy currently used by this builder.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[initialData](https://api.flutter.dev/flutter/widgets/StreamBuilder/initialData.html) → T?

The data that will be used to create the initial snapshot.

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[stream](https://api.flutter.dev/flutter/widgets/StreamBuilderBase/stream.html) → [Stream](https://api.flutter.dev/flutter/dart-async/Stream-class.html)<T>?

The asynchronous computation to which this builder is currently connected, possibly null. When changed, the current summary is updated using [afterDisconnected](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterDisconnected.html), if the previous stream was not null, followed by [afterConnected](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterConnected.html), if the new stream is not null.

finalinherited

### Methods

[afterConnected](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterConnected.html)([AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> current) → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns an updated version of the current summary reflecting that we are now connected to a stream.

override

[afterData](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterData.html)([AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> current, T data) → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns an updated version of the current summary following a data event.

override

[afterDisconnected](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterDisconnected.html)([AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> current) → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns an updated version of the current summary reflecting that we are no longer connected to a stream.

override

[afterDone](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterDone.html)([AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> current) → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns an updated version of the current summary following stream termination.

override

[afterError](https://api.flutter.dev/flutter/widgets/StreamBuilder/afterError.html)([AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> current, [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html) error, [StackTrace](https://api.flutter.dev/flutter/dart-core/StackTrace-class.html) stackTrace) → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns an updated version of the current summary following an error with a stack trace.

override

[build](https://api.flutter.dev/flutter/widgets/StreamBuilder/build.html)([BuildContext](https://api.flutter.dev/flutter/widgets/BuildContext-class.html) context, [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T> currentSummary) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

Returns a Widget based on the currentSummary.

override

[createElement](https://api.flutter.dev/flutter/widgets/StatefulWidget/createElement.html)() → [StatefulElement](https://api.flutter.dev/flutter/widgets/StatefulElement-class.html)

Creates a [StatefulElement](https://api.flutter.dev/flutter/widgets/StatefulElement-class.html) to manage this widget's location in the tree.

inherited

[createState](https://api.flutter.dev/flutter/widgets/StreamBuilderBase/createState.html)() → [State](https://api.flutter.dev/flutter/widgets/State-class.html)<[StreamBuilderBase](https://api.flutter.dev/flutter/widgets/StreamBuilderBase-class.html)<T, [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>>>

Creates the mutable state for this widget at a given location in the tree.

inherited

[debugDescribeChildren](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/debugDescribeChildren.html)() → [List](https://api.flutter.dev/flutter/dart-core/List-class.html)<[DiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticsNode-class.html)>

Returns a list of DiagnosticsNode objects describing this node's children.

inherited

[debugFillProperties](https://api.flutter.dev/flutter/widgets/Widget/debugFillProperties.html)([DiagnosticPropertiesBuilder](https://api.flutter.dev/flutter/foundation/DiagnosticPropertiesBuilder-class.html) properties) → void

Add additional properties associated with the node.

inherited

[initial](https://api.flutter.dev/flutter/widgets/StreamBuilder/initial.html)() → [AsyncSnapshot](https://api.flutter.dev/flutter/widgets/AsyncSnapshot-class.html)<T>

Returns the initial summary of stream interaction, typically representing the fact that no interaction has happened at all.

override

[noSuchMethod](https://api.flutter.dev/flutter/dart-core/Object/noSuchMethod.html)([Invocation](https://api.flutter.dev/flutter/dart-core/Invocation-class.html) invocation) → dynamic

Invoked when a nonexistent method or property is accessed.

inherited

[toDiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toDiagnosticsNode.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html)? name, [DiagnosticsTreeStyle](https://api.flutter.dev/flutter/foundation/DiagnosticsTreeStyle.html)? style}) → [DiagnosticsNode](https://api.flutter.dev/flutter/foundation/DiagnosticsNode-class.html)

Returns a debug representation of the object that is used by debugging tools and by [DiagnosticsNode.toStringDeep](https://api.flutter.dev/flutter/foundation/DiagnosticsNode/toStringDeep.html).

inherited

[toString](https://api.flutter.dev/flutter/foundation/Diagnosticable/toString.html)({[DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.info}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

A string representation of this object.

inherited

[toStringDeep](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toStringDeep.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html) prefixLineOne = '', [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? prefixOtherLines, [DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.debug}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

Returns a string representation of this node and its descendants.

inherited

[toStringShallow](https://api.flutter.dev/flutter/foundation/DiagnosticableTree/toStringShallow.html)({[String](https://api.flutter.dev/flutter/dart-core/String-class.html) joiner = ', ', [DiagnosticLevel](https://api.flutter.dev/flutter/foundation/DiagnosticLevel.html) minLevel = DiagnosticLevel.debug}) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

Returns a one-line detailed description of the object.

inherited

[toStringShort](https://api.flutter.dev/flutter/widgets/Widget/toStringShort.html)() → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

A short, textual description of this widget.

inherited

### Operators

[operator ==](https://api.flutter.dev/flutter/widgets/Widget/operator_equals.html)([Object](https://api.flutter.dev/flutter/dart-core/Object-class.html) other) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

The equality operator.

inherited