

**Module** `java.base`

**Package** `java.util.concurrent`

## Interface `StructuredTaskScope.Subtask<T>`

### Type Parameters:

T - the result type

### All Superinterfaces:

`Supplier<T>`

### Enclosing class:

`StructuredTaskScopePREVIEW<T>`

---

```
public static sealed interface StructuredTaskScope.Subtask<T>
    extends Supplier<T>
```

#### **Subtask is a preview API of the Java platform.**

*Programs can only use Subtask when preview features are enabled.*

*Preview features may be removed in a future release, or upgraded to permanent features of the Java platform.*

Represents a subtask forked with `StructuredTaskScope.fork(Callable)PREVIEW`.

### Since:

21

## ***Nested Class Summary***

### **Nested Classes**

Modifier and Type	Interface	Description

static enum

**StructuredTaskScope.Subtask.State**<sup>PREVIEW</sup> **Preview.**

Represents the state of a subtask.

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
Throwable	exception()	Returns the exception thrown by the subtask.
T	get()	Returns the result of the subtask.
StructuredTaskScope.Subtask.State <sup>PREVIEW</sup>	state()	Returns the state of the subtask.
Callable<? extends T>	task()	Returns the value returning task provided to the fork method.

Method Details

task
<pre>Callable&lt;? extends T&gt; task()</pre>
Returns the value returning task provided to the fork method.
<b>API Note:</b> Task objects with unique identity may be used for correlation by implementations of <code>handleComplete</code> <sup>PREVIEW</sup> .
<b>Returns:</b> the value returning task provided to the fork method

## state

`StructuredTaskScope.Subtask.StatePREVIEW state()`

Returns the state of the subtask.

**Returns:**

the state of the subtask

## get

`T get()`

Returns the result of the subtask.

To ensure correct usage, if the scope owner `forksPREVIEW` a subtask, then it must join (with `joinPREVIEW` or `joinUntilPREVIEW`) before it can obtain the result of the subtask.

**Specified by:**

`get` in interface `Supplier<T>`

**Returns:**

the possibly-null result

**Throws:**

`IllegalStateException` - if the subtask has not completed, did not complete successfully, or the current thread is the task scope owner and did not join after forking

**See Also:**

`StructuredTaskScope.Subtask.State.SUCCESSPREVIEW`

## exception

`Throwable exception()`

Returns the exception thrown by the subtask.

To ensure correct usage, if the scope owner `forksPREVIEW` a subtask, then it must join (with `joinPREVIEW` or `joinUntilPREVIEW`) before it can obtain the exception thrown by the subtask.

**Returns:**

the exception thrown by the subtask

**Throws:**

`IllegalStateException` - if the subtask has not completed, completed with a result, or the current thread is the task scope owner and did not join after forking

**See Also:**

`StructuredTaskScope.Subtask.State.FAILEDPREVIEW`

---

[Report a bug or suggest an enhancement](#)

For further API reference and developer documentation see the [Java SE Documentation](#), which contains more detailed, developer-targeted descriptions with conceptual overviews, definitions of terms, workarounds, and working code examples. [Other versions](#).

Java is a trademark or registered trademark of Oracle and/or its affiliates in the US and other countries.

Copyright © 1993, 2024, Oracle and/or its affiliates, 500 Oracle Parkway, Redwood Shores, CA 94065 USA.

All rights reserved. Use is subject to [license terms](#) and the [documentation redistribution policy](#). Modify [Cookie Preferences](#). Modify [Ad Choices](#).