

# `std::packaged_task`

`std::packaged_task` is a wrapper for a callable to be invoked asynchronously.

Dealing with `std::packaged_task` typically consists of four steps:

1. Wrap the work:

```
std::packaged_task<int(int, int)> task([](int a, int b){ return a + b; })
```

2. Create the future:

```
std::future<int> sum = task.get_future()
```

3. Perform the calculation:

```
task(2000, 11)
```

4. Query the result:

```
sum.get()
```

# std::packaged\_task

Member Functions	Description
<code>valid</code>	Checks if the task object has a valid function.
<code>get_future</code>	Returns the associated future.
<code>operator()</code>	Executes the function.
<code>make_ready_at_thread_exit</code>	Executes the function. The result is available if the current thread exits.
<code>reset</code>	Resets the state of the task. Abandons the stored results from previous executions.



A `std::packaged_task` can be in contrast to a `std::async` or a `std::promise` **reset and reused**.

`packagedTaskReuse.cpp`