





## CompletionService Interface

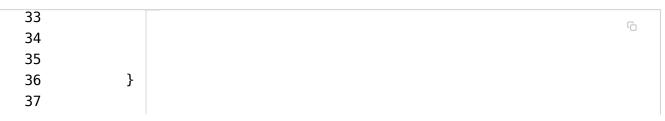
This lesson talks about how to batch multiple tasks together

## CompletionService Interface

In the previous lesson we discussed how tasks can be submitted to executors but imagine a scenario where you want to submit hundreds or thousands of tasks. You'll retrieve the future objects returned from the submit calls and then poll all of them in a loop to check which one is done and then take appropriate action. Java offers a better way to address this use case through the **CompletionService** interface. You can use the **ExecutorCompletionService** as a concrete implementation of the interface.

The completion service is a combination of a blocking queue and an executor. Tasks are submitted to the queue and then the queue can be polled for completed tasks. The service exposes two methods, one **poll** which returns null if no task is completed or none were submitted and two **take** which blocks till a completed task is available.

Below is an example program that demonstrates the use of completion service.



```
38
              Exec
39
              Exec
40
41
              // S
42
43
              for
44
              }
45
46
47
              // w
48
              int
49
              whil
50
51
52
53
54
              }
55
56
57
              thre
58
         }
59
60
```

Interviewing soon? We've partnered with Hired so that companies apply to yourm\_source=educative&utm\_medium=lesson&utm\_location=CA&utm\_campaignees

