

ExecutorService and Thread Pools



Maaike van Putten
Software Developer & Trainer

www.brightboost.nl



Overview



Future

ExecutorService

ScheduledExecutorService

Thread Pools

Submitting tasks



```
// Future objects
```

```
let website = {  
  name: 'Pluralsight',  
  url: 'https://pluralsight.com'  
};
```

```
function makeUrl() {}
```

```
let sites = []  
sites.push(website)
```

```
console.log(sites[0].name)
```

◀ **Create future object**

◀ **isDone()**

◀ **isCancelled()**

◀ **cancel()**

◀ **get()**

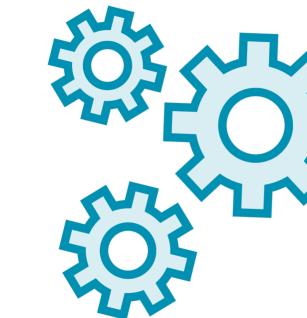
◀ **get(long time, TimeUnit unit)**

ExecutorService

ExecutorService is an interface in the `java.util.concurrent` package and helps us to manage tasks async.



Get an instance of ExecutorService



Execute tasks using ExecutorService



Close ExecutorService



Demo



Create an ExecutorService

Make it manage some tasks

Close ExecutorService



Submitting Tasks



Different methods

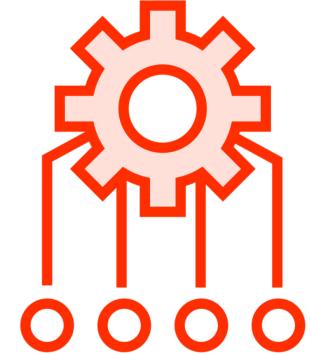
Methods depend on exact implementation of
ExecutorService interface

Tasks will be performed asynchronously

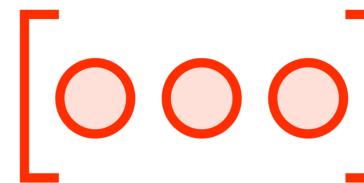
Afterwards, wait for ... the results



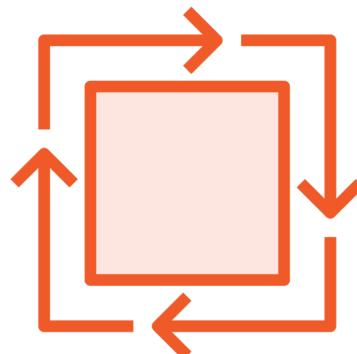
Thread Pool



Multiple threads that are performing tasks or waiting for a task. A bit like a worker crew!



Available threads pick up tasks from a queue.



When a thread is done, it will wait to pick up a new tasks instead of getting destroyed.





Different ExecutorServices

SingleThreadExecutor

SingleThreadScheduledExecutor

CachedThreadPool

FixedThreadPool

ScheduledThreadPool



Demo



**Executing tasks with different
ExecutorServices**

Examining the results



Demo



**Executing tasks with different
ScheduledExecutorServices**

Examining the results



ExecutorService: Overview of Methods

ExecutorService

- `execute`
- `submit`
- `invokeAll`
- `invokeAny`

ScheduledExecutorService

- `schedule`
- `scheduleAtFixedRate`
- `scheduleAtFixedDelay`



Up Next:
Concurrent Collections and Atomic Variables

