Executor Service

- Executor for Runnable
- Callable
- java.util.concurrent.Future
- ExecutorService
- ScheduledExecutorService
- Examples

java.util.concurrent.Executor

Executes the given command at some time in the future.

The command may execute in a new thread, in a pooled thread, or in the calling thread, at the discretion of the Executor implementation.

```
public interface Executor {
    void execute(Runnable command);
}
```

DirectExecutor

```
class DirectExecutor implements Executor {
        public void execute(Runnable r) {
                r.run();
        }
}
```

ThreadPerTaskExecutor

```
class ThreadPerTaskExecutor implements Executor {
        public void execute(Runnable r) {
                new Thread(r).start();
        }
}
```

java.util.concurrent.Callable

The Callable interface is similar to java.lang.Runnable, in that both are designed for classes whose instances are potentially executed by another thread.

A Runnable, however, does not return a result and cannot throw a checked exception.

```
@FunctionalInterface
public interface Callable<V> {
    V call() throws Exception;
}
```

java.util.concurrent.Future

A Future represents the result of an asynchronous computation.

The result can only be retrieved using method get() when the computation has completed, blocking if necessary until it is ready.

```
public interface Future<V> {
    boolean cancel(boolean mayInterruptIfRunning);
    boolean isCancelled();
    boolean isDone();
    V get() throws InterruptedException, ExecutionException;
    V get(long timeout, TimeUnit unit)
        throws InterruptedException, ExecutionException, TimeoutException;
}
```

ExecutorService

```
public interface ExecutorService extends Executor
```

An Executor that provides methods to manage creation of new threads, termination of threads and methods that can produce a Future for tracking progress of one or more asynchronous tasks.

Method submit() extends base method execute(Runnable) by creating and returning a Future that can be used to cancel execution and/or wait for completion.

ExecutorService pool = Executors.newFixedThreadPool(10);

Callable<Integer> callable = new TaskExtendedCallable();

Future<Integer> future = pool.submit(callable);

ScheduledExecutorService

An ExecutorService that can schedule commands to run after a given delay, or to execute periodically.