The Java ExecutorService Interface

(Part 1)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Recognize the powerful features defined in the Java ExecutorService interface & related interfaces/classes
Overview of the ExecutorService Interface
Overview of the ExecutorService Interface

- Extends Executor

```
<<Java Interface>>

ExecutorService

- shutdown():void
- shutdownNow():List<Runnable>
- isShutdown():boolean
- isTerminated():boolean
- awaitTermination(long, TimeUnit):boolean
- submit(Callable<T>):Future<T>
- submit(Runnable,T):Future<T>
- submit(Runnable):Future<?>
- invokeAll(Collection<? extends Callable<T>>):List<Future<T>>
- invokeAny(Collection<? extends Callable<T>>)  
- invokeAny(Collection<? extends Callable<T>>,long, TimeUnit)
```

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html)
Overview of the ExecutorService Interface

- Extends Executor
- Submit tasks & return futures for these tasks

```java
<<Java Interface>>
ExecutorService

shutdown():void
shutdownNow():List<Runnable>
isShutdown():boolean
isTerminated():boolean
awaitTermination(long, TimeUnit):boolean
submit(Callable<T>):Future<T>
submit(Runnable,T):Future<T>
submit(Runnable):Future<?>
invokeAll(Collection<? extends Callable<T>>):List<Future<T>>
invokeAny(Collection<? extends Callable<T>>)
invokeAny(Collection<? extends Callable<T>>,long,TimeUnit)
```

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html
Overview of the ExecutorService Interface

- Extends Executor
- Submit tasks & return futures for these tasks
- Manage lifecycle of tasks & worker threads

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/ExecutorService.html)
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces

<<Java Interface>>

Future<V>

- cancel(boolean):boolean
- isCancelled():boolean
- isDone():boolean
- get()
- get(long, TimeUnit)

<<Java Interface>>

Callable<V>

- call()

<<Java Interface>>

Runnable

- run():void
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - **Runnable**
    - A “one-way” task that does not return a result

See [docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html](http://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html)
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
    - A “two-way” task that returns a result

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/Callable.html)
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
    - A “two-way” task that returns a result
    - Typically used to run two-way async tasks

<<Java Interface>>

Callable<V>

• call()
The ExecutorService is used with other interfaces, e.g.

- **Runnable**
- **Callable**
  - A “two-way” task that returns a result
  - Typically used to run two-way async tasks
- Implements the *Active Object* pattern

Decouples the thread that invokes a method from the thread that executes the method

See [en.wikipedia.org/wiki/Active_object](https://en.wikipedia.org/wiki/Active_object)
Overview of the `ExecutorService` Interface

- The `ExecutorService` is used with other interfaces, e.g.
  - `Runnable`
  - `Callable`
  - `Future`
    - Represents async two-way task result

---

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/Future.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/Future.html)
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
  - Future
    - Represents async two-way task result
      - Can be canceled & tested for completion

```
<<Java Interface>>
Future<V>
- cancel(boolean):boolean
- isCancelled():boolean
- isDone():boolean
- get()
- get(long, TimeUnit)
```
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
- Future
  - Represents async two-way task result
    - Can be canceled & tested for completion
  - Can implement Active Object pattern

See en.wikipedia.org/wiki/Active_object
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
- Future
  - Represents async two-way task result
    - Can be canceled & tested for completion
    - Can implement *Active Object* pattern
  - Other Future variants implement ExecutorService
Overview of the `ExecutorService` Interface

- The `ExecutorService` is used with other interfaces, e.g.
  - `Runnable`
  - `Callable`
- `Future`
  - Represents async two-way task result
    - Can be canceled & tested for completion
    - Can implement *Active Object* pattern
  - Other Future variants implement `ExecutorService`
    - `FutureTask` – Conveys result from thread running a computation to thread(s) retrieving result

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/FutureTask.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/FutureTask.html)
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
- Future
  - Represents async two-way task result
    - Can be canceled & tested for completion
    - Can implement Active Object pattern
  - Other Future variants implement ExecutorService
    - FutureTask
      - RunnableFuture – Execution of run() method will complete the future & allow access to its results

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/RunnableFuture.html
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
  - Future
  - CompletableFuture

- This Future variant supports dependent actions that are trigger upon its completion

CompletableFuture isn’t part of the Java Executor framework

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html
Overview of the ExecutorService Interface

- The ExecutorService is used with other interfaces, e.g.
  - Runnable
  - Callable
  - Future
  - CompletableFuture
    - This Future variant supports dependent actions that are triggered upon its completion
    - This topic is covered in other courses

See [www.dre.vanderbilt.edu/~schmidt/DigitalLearning](http://www.dre.vanderbilt.edu/~schmidt/DigitalLearning)
Overview of the ExecutorService Interface

- ExecutorService also forms the basis for key Java Executor framework subclasses

See src/share/classes/java/util/concurrent
End of Overview of Java ExecutorService (Part 1)