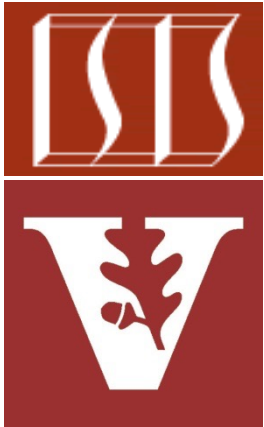


Overview of the OneShot ExecutorCompletionService Class













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

**Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Understand the SearchTaskGang case study
- Recognize the methods that are defined by the TaskGang framework
- Know the subclasses that extends TaskGang (directly or indirectly)
 - SearchTaskGangCommon
 - OneShotThreadPerTask
 - OneShotExecutorService
 - OneShotExecutorServiceFuture
 - OneShotExecutorCompletionService

		OneShotExecutorCompletionService
		OneShotExecutorCompletionService(String[], String[][])
		mCompletionService ExecutorCompletionService<SearchResults>
		concurrentlyProcessQueuedFutures() void
		initiateTaskGang(int) void

See [SearchTaskGang/src/main/java/tasks/OneShotExecutorCompletionService.java](#)

Overview of OneShotExecutor CompletionService

Overview of OneShotExecutorCompletionService

TaskGang<E>	
m	TaskGang()
m	advanceTaskToNextCycle() boolean
m	awaitTasksDone() void
m	currentCycle() long
m	incrementCycle() long
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	makeTask(int) Runnable
m	processInput(E) boolean
m	run() void
m	setInput(List<E>) List<E>
m	taskDone(int) void
p	executor Executor
p	input List<E>
p	nextInput List<E>

SearchTaskGangTest	
m	SearchTaskGangTest()
m	main(String[]) void
m	makeTaskGang(String[], TestsToRun) Runnable

SearchTaskGangCommon	
m	SearchTaskGangCommon(String[], String[][]) void
m	awaitTasksDone() void
m	makeResultsIndexOf(String, String) List<Result>
m	searchForWord(String, String) SearchResults
p	nextInput List<String>

OneShotExecutorService	
m	OneShotExecutorService(String[], String[][]) void
m	awaitTasksDone() void
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	processInput(String) boolean
m	processQueuedResults(int) void
m	taskDone(int) void

OneShotThreadPerTask	
m	OneShotThreadPerTask(String[], String[][]) void
m	awaitTasksDone() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][]) void
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

OneShotExecutorServiceFuture	
m	OneShotExecutorServiceFuture(String[], String[][]) void
m	initiateTaskGang(int) void
m	processFutureResults(List<Future<SearchResults>>) void
m	processInput(String) boolean

Customizes SearchTaskGang Common to do one-shot search for words in a List of Strings with maximal concurrency

Overview of OneShotExecutorCompletionService

TaskGang<E>	
m	TaskGang()
m	advanceTaskToNextCycle() boolean
m	awaitTasksDone() void
m	currentCycle() long
m	incrementCycle() long
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	makeTask(int) Runnable
m	processInput(E) boolean
m	run() void
m	setInput(List<E>) List<E>
m	taskDone(int) void
p	executor Executor
p	input List<E>
p	nextInput List<E>

SearchTaskGangTest	
m	SearchTaskGangTest()
m	main(String[]) void
m	makeTaskGang(String[], TestsToRun) Runnable

SearchTaskGangCommon	
m	SearchTaskGangCommon(String[], String[][]) void
m	awaitTasksDone() void
m	makeResultsIndexOf(String, String) List<Result>
m	searchForWord(String, String) SearchResults
p	nextInput List<String>

OneShotExecutorService	
m	OneShotExecutorService(String[], String[][]) void
m	awaitTasksDone() void
m	initiateHook(int) void
m	initiateTaskGang(int) void
m	processInput(String) boolean
m	processQueuedResults(int) void
m	taskDone(int) void

OneShotThreadPerTask	
m	OneShotThreadPerTask(String[], String[][]) void
m	awaitTasksDone() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][]) void
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void
m	processInput(String) boolean

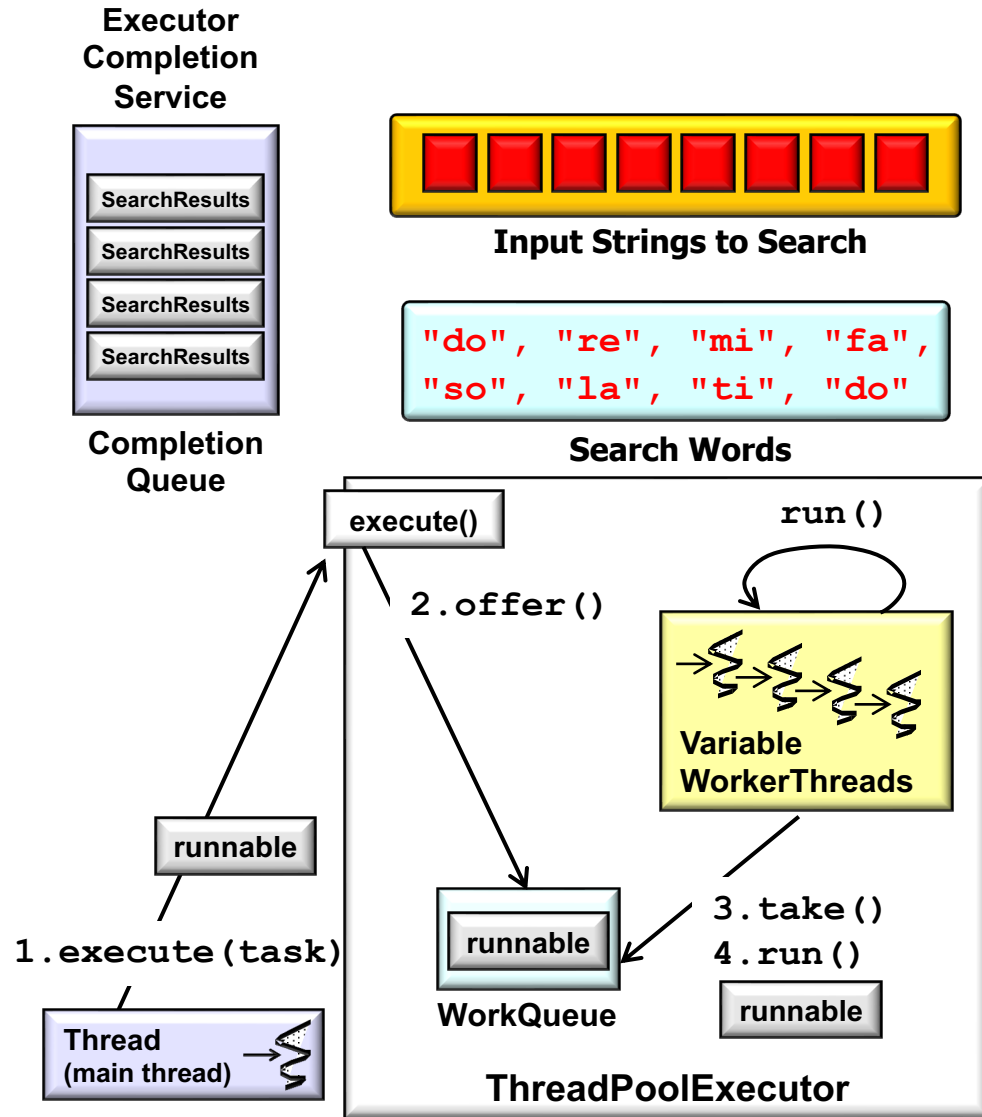
OneShotExecutorServiceFuture	
m	OneShotExecutorServiceFuture(String[], String[][]) void
m	initiateTaskGang(int) void
m	processFutureResults(List<Future<SearchResults>>) void
m	processInput(String) boolean

See [SearchTaskGang/src/main/java/tasks/OneShotExecutorCompletionService.java](https://github.com/OneShot/OneShotExecutorCompletionService/blob/master/src/main/java/tasks/OneShotExecutorCompletionService.java)

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
- Common to search for words in a List of String objects with maximal concurrency

Strategy	Implementation
<i>Executor model</i>	"Work-stealing" Thread pool
<i>Unit of concurrency</i>	Task per search word & input String
<i>Results processing model</i>	Asynchronous Future model processes results immediately



Overcomes limitations with the synchronous Future processing model via `ExecutorCompletionService`

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
Common to search for words
in a List of String objects with
maximal concurrency

OneShotExecutorCompletionService	
m	<code>OneShotExecutorCompletionService(String[], String[])</code>
f	<code>mCompletionService</code> ExecutorCompletionService<SearchResults>
m	<code>concurrentlyProcessQueuedFutures()</code> void
m	<code>initiateTaskGang(int)</code> void

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
Common to search for words in a List of String objects with maximal concurrency
- Uses a “work-stealing” Thread pool
 - Created by an Executors factory method

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][])
f	mCompletionService ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void

```
setExecutor  
    (Executors.newWorkStealingPool());  
  
mCompletionService = new  
    ExecutorCompletionService<>  
    (getExecutor());
```


Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
 - Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Concurrently process search results as Futures are done

OneShotExecutorCompletionService		
m	OneShotExecutorCompletionService(String[], String[])	
f	mCompletionService	ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures()	void
m	initiateTaskGang(int)	void

```
setExecutor
    (Executors.newWorkStealingPool());

mCompletionService = new
    ExecutorCompletionService<>
        (getExecutor());
```

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently

OneShotExecutorCompletionService		
m	OneShotExecutorCompletionService(String[], String[][])	
f	mCompletionService ExecutorCompletionService<SearchResults>	
m	concurrentlyProcessQueuedFutures()	void
m	initiateTaskGang(int)	void

```
for (int i = 0; i < inputSize; ++i)
    getExecutorService().
        execute(makeTask(i));

concurrentlyProcessQueuedFutures();
```

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
 - Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently
 - Results processing starts once any Future is done

OneShotExecutorCompletionService		
m	OneShotExecutorCompletionService(String[], String[][])	
f	mCompletionService ExecutorCompletionService<SearchResults>	
m	concurrentlyProcessQueuedFutures()	void
m	initiateTaskGang(int)	void

```
for (int i = 0; i < inputSize; ++i)
    getExecutorService().
        execute(makeTask(i));
concurrentlyProcessQueuedFutures();
```

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently
 - Uses ExecutorService to search each word in input concurrently

OneShotExecutorCompletionService		
m	OneShotExecutorCompletionService(String[], String[][])	
f	mCompletionService	ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures()	void
m	initiateTaskGang(int)	void

```
for (var word : mWordsToFind) {  
    mCompletionService().submit  
        (( () -> searchForWord(word,  
                                inputData) ));  
}
```

The Future objects returned by submit() are ignored here

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
 - Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently
 - Uses ExecutorService to search each word in input concurrently
 - Future results are processed concurrently wrt submit() calls

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][])
f	mCompletionService ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void

```
for (int i = 0; i < count; ++i) {
    var resultFuture =
        rethrowSupplier
            (mCompletionService::take)
            .get();

    resultFuture.resultNow().print();
}
```

Overview of OneShotExecutorCompletionService

- Customizes SearchTaskGang
 - Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently
 - Uses ExecutorService to search each word in input concurrently
 - Future results are processed concurrently wrt submit() calls
 - take() may block

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][])
f	mCompletionService ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void

```
for (int i = 0; i < count; ++i) {
    var resultFuture =
        rethrowSupplier
            (mCompletionService::take)
            .get();

    resultFuture.resultNow().print();
}
```

Overview of OneShotExecutorCompletionService

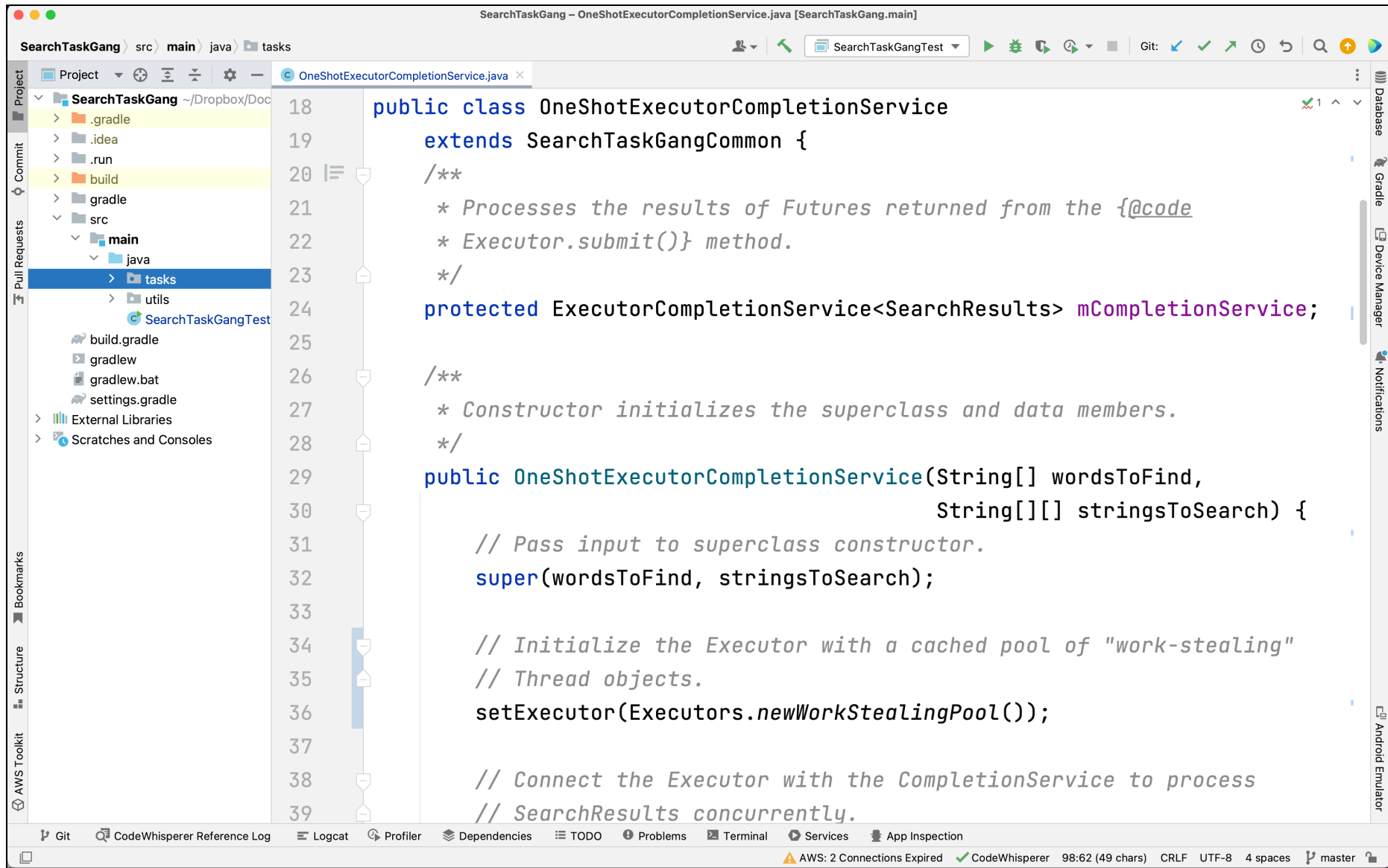
- Customizes SearchTaskGang
 - Common to search for words in a List of String objects with maximal concurrency
 - Uses a cached Thread pool
 - Connect the Executor with ExecutorCompletionService
 - Each input data element is processed concurrently
 - Uses ExecutorService to search each word in input concurrently
 - Future results are processed concurrently wrt submit() calls
 - take() may block
 - resultNow() doesn't block since Futures aren't queued until they're done

OneShotExecutorCompletionService	
m	OneShotExecutorCompletionService(String[], String[][])
f	mCompletionService ExecutorCompletionService<SearchResults>
m	concurrentlyProcessQueuedFutures() void
m	initiateTaskGang(int) void

```
for (int i = 0; i < count; ++i) {
    var resultFuture =
        rethrowSupplier
            (mCompletionService::take)
            .get();

    resultFuture.resultNow().print();
}
```

Walkthrough of OneShotExecutorCompletionService



```
SearchTaskGang – OneShotExecutorCompletionService.java [SearchTaskGang.main]
SearchTaskGang \ src \ main \ java \ tasks
Project SearchTaskGang ~/Dropbox/Doc
  .gradle
  .idea
  .run
  build
  gradle
  src
    main
      java
        tasks
          SearchTaskGangTest
        build.gradle
        gradlew
        gradlew.bat
        settings.gradle
      External Libraries
      Scratches and Consoles
Commit
Pull Requests
Bookmarks
Structure
AWS Toolkit
Database
Gradle
Device Manager
Notifications
Android Emulator

18 public class OneShotExecutorCompletionService
19 extends SearchTaskGangCommon {
20 /**
21  * Processes the results of Futures returned from the {@code
22  * Executor.submit()} method.
23  */
24 protected ExecutorCompletionService<SearchResults> mCompletionService;
25
26 /**
27  * Constructor initializes the superclass and data members.
28  */
29 public OneShotExecutorCompletionService(String[] wordsToFind,
30                                         String[][] stringsToSearch) {
31     // Pass input to superclass constructor.
32     super(wordsToFind, stringsToSearch);
33
34     // Initialize the Executor with a cached pool of "work-stealing"
35     // Thread objects.
36     setExecutor(Executors.newWorkStealingPool());
37
38     // Connect the Executor with the CompletionService to process
39     // SearchResults concurrently.
```

Git CodeWhisperer Reference Log Logcat Profiler Dependencies TODO Problems Terminal Services App Inspection

AWS: 2 Connections Expired CodeWhisperer 98:62 (49 chars) CRLF UTF-8 4 spaces master

See [SearchTaskGang/src/main/java/tasks/OneShotExecutorCompletionService.java](https://github.com/SearchTaskGang/src/main/java/tasks/OneShotExecutorCompletionService.java)

End of Overview of OneShot ExecutorCompletionService