Overview of Java Futures

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

- Motivate the need for Java futures by understanding the pros & cons of synchrony & asynchrony.
- Understand that Java futures provide the foundation for completable futures in Java.

See [en.wikipedia.org/wiki/Java_version_history](en.wikipedia.org/wiki/Java_version_history)
Learning Objectives in this Part of the Lesson

- Motivate the need for Java futures by understanding the pros & cons of synchrony & asynchrony
- Understand that Java futures provide the foundation for completable futures in Java
- Recognize a human known use of Java futures
Learning Objectives in this Part of the Lesson

• Motivate the need for Java futures by understanding the pros & cons of synchrony & asynchrony

• Understand that Java futures provide the foundation for completable futures in Java
  • Recognize a human known use of Java futures
  • Know all the methods in the Future interface

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/Future.html
A Human Known Use of Java Futures
A Human Known Use of Java Futures

- A future is essentially a proxy that represents the result(s) of an async call.

1. Async computation runs
2. Result obtained only after async computation completes

See en.wikipedia.org/wiki/Futures_and_promises
A Human Known Use of Java Futures

- Table tent #'s & table # stands a human-known-use of futures in restaurants!

See www.citygrafx.com/table-numbers-table-markers
A Human Known Use of Java Futures

- Table tent #’s & table # stands a human-known-use of futures in restaurants!
- e.g., McDonald’s vs Wendy’s model of preparing fast food
A Human Known Use of Java Futures

- Table tent #'s & table # stands a human-known-use of futures in restaurants!
- e.g., McDonald’s vs Wendy’s model of preparing fast food


McDonald’s historically ‘cached’ food in heatlamps & performed “synchronous” transactions
A Human Known Use of Java Futures

- Table tent #’s & table # stands a human-known-use of futures in restaurants!
- e.g., McDonald’s vs Wendy’s model of preparing fast food

Wendy’s historically cooked food to order & performed “asynchronous” transactions with various futures

See www.wendys.com/csr-what-we-value/food/quality/fresh
Overview of the Java Future API
Overview of the Java Future API

- Java 5 added async call support via the Java Future interface

See en.wikipedia.org/wiki/Java_version_history
Overview of the Java Future API

- Java Future methods can manage a task’s lifecycle after it’s submitted to run asynchronously.

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/Future.html
Overview of the Java Future API

- Java Future methods can manage a task’s lifecycle after it’s submitted to run asynchronously, e.g.
  - A future can be tested for completion

```
1. submit(task)
2. Return future
3. offer()
4. take()
5. run()
6. isDone()
```

```java
Future<V> future = executor.submit(new Callable<V>() {
    @Override
    public V call() throws Exception {
        // Task logic
    }
});

if (future.isDone()) {
    // Task completed
}
```

- `Future<V>`:
  - `cancel(boolean)`: boolean
  - `isCancelled()`: boolean
  - `isDone()`: boolean
  - `get()`: V
  - `get(long, TimeUnit)`: V

Diagram:
- ThreadPoolExecutor
- WorkQueue
- Callable
- Fixed WorkerThreads
- Run Method
- Return Future Method
- Submit Method
Overview of the Java Future API

- Java Future methods can manage a task’s lifecycle after it’s submitted to run asynchronously, e.g.
  - A future can be tested for completion
  - A future be tested for cancellation & cancelled

```
Future future = executor.submit(task);
if (!future.isCancelled())
    future.cancel();
```

```
Callable callable = new Callable()
Fixed WorkerThreads
```
Overview of the Java Future API

- Java Future methods can manage a task’s lifecycle after it’s submitted to run asynchronously, e.g.
  - A future can be tested for completion
  - A future be tested for cancellation & cancelled
  - A future can retrieve a two-way task’s result
Overview of the Java Future API

- The Java Future interface provides the foundation for the Java CompletableFuture class

See en.wikipedia.org/wiki/Java_version_history
Overview of the Java Future API

- The Java Future interface provides the foundation for the Java CompletableFuture class.
- However, the CompletableFuture class defines dozens of methods & more powerful capabilities.
End of Overview of Java Futures