Understand Enhancements to the Java Completable Futures Framework

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

• Evaluate the pros of using the Java completable futures framework
• Evaluate the cons of using the Java completable futures framework
• Understand enhancements to the Java completable futures framework
Enhancements to the Java Completable Futures Framework
Enhancements to the Java Completable Futures Framework

- Java 9 provides enhancements to the Java 8 completable future framework

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default.Executor</td>
<td>()</td>
<td>Returns default <code>Executor</code> used for methods that don’t specify an <code>Executor</code></td>
</tr>
<tr>
<td>complete.Async</td>
<td>Supplier&lt;T&gt;</td>
<td>Complete <code>CompletableFuture</code> asynchronously using value given by the <code>Supplier</code></td>
</tr>
<tr>
<td>orTimeout</td>
<td>long timeout, TimeUnit unit</td>
<td>Resolves <code>CompletableFuture</code> exceptionally with <code>TimeoutException</code>, unless it is completed before the specified timeout</td>
</tr>
<tr>
<td>complete.OnTimeout</td>
<td>T value, long timeout, TimeUnit unit</td>
<td>Completes this <code>CompletableFuture</code> with the given value if not otherwise completed before the given timeout</td>
</tr>
</tbody>
</table>

See [www.baeldung.com/java-9-completablefuture](http://www.baeldung.com/java-9-completablefuture)
Enhancements to the Java Completable Futures Framework

Java 9 provides enhancements to the Java 8 completable future framework.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>Returns or Completes</th>
</tr>
</thead>
<tbody>
<tr>
<td>default Executor ()</td>
<td>Executor</td>
<td>default Executor used for methods that don’t specify an Executor</td>
</tr>
<tr>
<td>complete Async Supplier&lt;T&gt;</td>
<td>CompletableObjectFuture&lt;T&gt;</td>
<td>Complete CompletableFuture asynchronously using value given by the Supplier</td>
</tr>
<tr>
<td>orTimeout long timeout, TimeUnit unit</td>
<td>CompletableObjectFuture&lt;T&gt;</td>
<td>Resolves CompletableFuture exceptionally with TimeoutException, unless it is completed before the specified timeout</td>
</tr>
<tr>
<td>complete OnTimeout T value, long timeout, TimeUnit unit</td>
<td>CompletableObjectFuture&lt;T&gt;</td>
<td>Completes this CompletableFuture with the given value if not otherwise completed before the given timeout</td>
</tr>
</tbody>
</table>

See [docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#defaultExecutor](http://docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#defaultExecutor)
Enhancements to the Java Completable Futures Framework

- Java 9 provides enhancements to the Java 8 completable future framework

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>default</code></td>
<td><code>Executor</code></td>
<td>Returns default <code>Executor</code> used for methods that don’t specify an <code>Executor</code></td>
</tr>
<tr>
<td><code>complete</code></td>
<td><code>Supplier&lt;T&gt;</code></td>
<td>Complete <code>CompletableFuture</code> asynchronously using value given by the <code>Supplier</code></td>
</tr>
<tr>
<td><code>orTimeout</code></td>
<td><code>long timeout, TimeUnit unit</code></td>
<td>Resolves <code>CompletableFuture</code> exceptionally with <code>TimeoutException</code>, unless it is completed before the specified timeout</td>
</tr>
<tr>
<td><code>complete</code></td>
<td><code>T value, long timeout, TimeUnit unit</code></td>
<td>Completes this <code>CompletableFuture</code> with the given value if not otherwise completed before the given timeout</td>
</tr>
</tbody>
</table>

See [docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#completeAsync](https://docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#completeAsync)
Enhancements to the Java Completable Futures Framework

- Java 9 provides enhancements to the Java 8 completable future framework

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>()</td>
<td>Executor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns default <code>Executor</code> used for methods that don’t specify an <code>Executor</code></td>
</tr>
<tr>
<td>complete</td>
<td>Supplier&lt;T&gt;</td>
<td>CompletableObject&lt;T&gt;</td>
</tr>
<tr>
<td>Async</td>
<td></td>
<td>Complete <code>CompletableObject</code> asynchronously using value given by the <code>Supplier</code></td>
</tr>
<tr>
<td>orTimeout</td>
<td>long timeout, TimeUnit unit</td>
<td>CompletableObject&lt;T&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resolves <code>CompletableObject</code> exceptionally with <code>TimeoutException</code>, unless it is completed before the specified timeout</td>
</tr>
<tr>
<td>complete</td>
<td>T value, long timeout, TimeUnit unit</td>
<td>CompletableObject&lt;T&gt;</td>
</tr>
<tr>
<td>OnTimeout</td>
<td></td>
<td>Completes this <code>CompletableObject</code> with the given value if not otherwise completed before the given timeout</td>
</tr>
</tbody>
</table>

See [docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#orTimeout](http://docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#orTimeout)
Enhancements to the Java Completable Futures Framework

- Java 9 provides enhancements to the Java 8 completable future framework

<table>
<thead>
<tr>
<th>Methods</th>
<th>Params</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default Executor ()</td>
<td>Executor</td>
<td>Returns default Executor used for methods that don’t specify an Executor</td>
</tr>
<tr>
<td>complete Async Supplier&lt;T&gt;</td>
<td>CompletableFuture&lt;T&gt;</td>
<td>Complete CompletableFuture asynchronously using value given by the Supplier</td>
</tr>
<tr>
<td>orTimeout long timeout, TimeUnit unit</td>
<td>CompletableFuture&lt;T&gt;</td>
<td>Resolves CompletableFuture exceptionally with TimeoutException, unless it is completed before the specified timeout</td>
</tr>
<tr>
<td>complete OnTimeout T value, long timeout, TimeUnit unit</td>
<td>CompletableFuture&lt;T&gt;</td>
<td>Completes this CompletableFuture with the given value if not otherwise completed before the given timeout</td>
</tr>
</tbody>
</table>

See docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#completeOnTimeout
Applying orTimeout() & completeOnTimeout()
Applying the `orTimeout()` & `completeOnTimeout()` Methods

- This example asynchronously performs web services within a bounded time

```java
CompletableFuture.supplyAsync()
  .thenCombine(CompletableFuture.supplyAsync()
    .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
    this::convert)
  .orTimeout(3, SECONDS)
  .whenComplete((amount, ex) -> {
    if (amount != null)
      { System.out.println("The price is: " + amount + "GBP"); } 
    else { System.out.println(ex.toString()); } 
  })...
Applying the orTimeout() & completeOnTimeout() Methods

- This example asynchronously performs web services within a bounded time

```java
CompletableFuture.supplyAsync()
    .thenCombine(CompletableFuture.supplyAsync
        .supplyAsync(() -> findBestPrice("LDN - NYC"))
        .thenCombine(CompletableFuture.supplyAsync(() -> queryExchangeRateFor("GBP"))
            .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
            this::convert)
        .orTimeout(3, SECONDS)
    .thenComplete((amount, ex) -> {
        if (amount != null)
            System.out.println("The price is: " + amount + "GBP");
        else System.out.println(ex.toString());
    }...)
```

supplyAsync() uses the default executor (i.e., common fork-join pool)

Asynchronously find the best price for a flight
Applying the orTimeout() & completeOnTimeout() Methods

• This example asynchronously performs web services within a bounded time

```java
CompletableFuture
  .supplyAsync()
    () -> findBestPrice("LDN - NYC")
  .thenCombine(CompletableFuture
    .supplyAsync
      () -> queryExchangeRateFor("GBP")
    .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
      this::convert)
  .orTimeout(3, SECONDS)
  .whenComplete((amount, ex) -> {
    if (amount != null)
      { System.out.println("The price is: " + amount + "GBP"); } 
    else { System.out.println(ex.toString()); } 
  })...
```

Asynchronously find the latest exchange rate
Applying the `orTimeout()` & `completeOnTimeout()` Methods

- This example asynchronously performs web services within a bounded time

```java
CompletableFuture
    .supplyAsync(
        () -> findBestPrice("LDN - NYC"))
    .thenCombine(CompletableFuture
        .supplyAsync
            (() -> queryExchangeRateFor("GBP"))
        .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
            this::convert)
    .orTimeout(3, SECONDS)
    .whenComplete((amount, ex) -> {
        if (amount != null)
            { System.out.println("The price is: " + amount + "GBP"); }
        else { System.out.println(ex.toString()); } 
    })...
```

Select the default rate if call runs longer than 2 seconds

See iteratrlearning.com/java9/2016/09/13/java9-timeouts-completablefutures.html
Applying the `orTimeout()` & `completeOnTimeout()` Methods

- This example asynchronously performs web services within a bounded time

```java
CompletableFuture
    .supplyAsync(
        () -> findBestPrice("LDN - NYC"))
    .thenCombine(CompletableFuture
        .supplyAsync
            (() -> queryExchangeRateFor("GBP"))
        .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
        this::convert)
    .orTimeout(3, SECONDS)
    .whenComplete((amount, ex) -> {
        if (amount != null)
            { System.out.println("The price is: " + amount + "GBP"); }
        else { System.out.println(ex.toString()); }
    })...
```

Combine & convert the search results
CompletableFuture
  .supplyAsync()
  .thenCombine(CompletableFuture
    .supplyAsync()
    .thenCombine(CompletableFuture
      .supplyAsync()
      .thenCombine(CompletableFuture
        .supplyAsync()
        .thenCombine(CompletableFuture
          .supplyAsync()
          .thenCombine(CompletableFuture
            .supplyAsync()
            .thenCombine(CompletableFuture
              .supplyAsync()
              .thenCombine(CompletableFuture
                .supplyAsync()
                .thenCombine(CompletableFuture
                  .supplyAsync()
                  .thenCombine(CompletableFuture
                    .supplyAsync()
                    .thenCombine(CompletableFuture
                      .supplyAsync()
                      .thenCombine(CompletableFuture
                        .supplyAsync()
                        .thenCombine(CompletableFuture
                          .supplyAsync()
                          .thenCombine(CompletableFuture
                            .supplyAsync()
                            .thenCombine(CompletableFuture
                              .supplyAsync()
                              .thenCombine(CompletableFuture
                                .supplyAsync()
                                .thenCombine(CompletableFuture
                                  .supplyAsync()
                                  .thenCombine(CompletableFuture
                                    .supplyAsync()
                                    .thenCombine(CompletableFuture
                                      .supplyAsync()
                                      .thenCombine(CompletableFuture
                                        .supplyAsync()
                                        .thenCombine(CompletableFuture
                                          .supplyAsync()
                                          .thenCombine(CompletableFuture
                                            .supplyAsync()
                                            .thenCombine(CompletableFuture
                                              .supplyAsync()
                                              .thenCombine(CompletableFuture
                                                .supplyAsync()
                                                .thenCombine(CompletableFuture
                                                  .supplyAsync()
                                                  .thenCombine(CompletableFuture
                                                    .supplyAsync()
                                                    .thenCombine(CompletableFuture
                                                      .supplyAsync()
                                                      .thenCombine(CompletableFuture
                                                        .supplyAsync()
                                                        .thenCombine(CompletableFuture
                                                          .supplyAsync()
                                                          .thenCombine(CompletableFuture
                                                            .supplyAsync()
                                                            .thenCombine(CompletableFuture
                                                              .supplyAsync()
                                                              .thenCombine(CompletableFuture
                                                                .supplyAsync()
                                                                .thenCombine(CompletableFuture
                                                                  .supplyAsync()
                                                                  .thenCombine(CompletableFuture
                                                                    .supplyAsync()
                                                                    .thenCombine(CompletableFuture
                                                                      .supplyAsync()
                                                                      .thenCombine(CompletableFuture
                                                                        .supplyAsync()
                                                                        .thenCombine(CompletableFuture
                                                                          .supplyAsync()
                                                                          .thenCombine(CompletableFuture
                                                                            .supplyAsync()
                                                                            .thenCombine(CompletableFuture
                                                                              .supplyAsync()
                                                                              .thenCombine(CompletableFuture
                                                                                .supplyAsync()
                                                                                .thenCombine(CompletableFuture
                                                                                  .supplyAsync()
                                                                                  .thenCombine(CompletableFuture
                                                                                    .supplyAsync()
                                                                                    .thenCombine(CompletableFuture
                                                                                      .supplyAsync()
                                                                                      .thenCombine(CompletableFuture
                                                                                       .supplyAsync()
                                                                                       .thenCombine(CompletableFuture
                                                                                        .supplyAsync()
                                                                                        .thenCombine(CompletableFuture
                                                                                          .supplyAsync()
                                                                                          .thenCombine(CompletableFuture
                                                                                           .supplyAsync()
                                                                                           .thenCombine(CompletableFuture
                                                                                             .supplyAsync()
                                                                                             .thenCombine(CompletableFuture
                                                                                              .supplyAsync()
                                                                                              .thenCombine(CompletableFuture
                                                                                               .supplyAsync()
                                                                                               .thenCombine(CompletableFuture
                                                                                                 .supplyAsync()
                                                                                                 .thenCombine(CompletableFuture
                                                                                                  .supplyAsync()
                                                                                                  .thenCombine(CompletableFuture
                                                                                                   .supplyAsync()
                                                                                                   .thenCombine(CompletableFuture
                                                                                                    .supplyAsync()
                                                                                                    .thenCombine(CompletableFuture
                                                                                                     .supplyAsync()
                                                                                                     .thenCombine(CompletableFuture
                                                                                                      .supplyAsync()
                                                                                                      .thenCombine(CompletableFuture
                                                                                                       .supplyAsync()
                                                                                                       .thenCombine(CompletableFuture
                                                                                                        .supplyAsync()
                                                                                                        .thenCombine(CompletableFuture
                                                                                                         .supplyAsync()
                                                                                                         .thenCombine(CompletableFuture
                                                                                                          .supplyAsync()
                                                                                                          .thenCombine(CompletableFuture
                                                                                                           .supplyAsync()
                                                                                                           .thenCombine(CompletableFuture
                                                                                                            .supplyAsync()
                                                                                                            .thenCombine(CompletableFuture
                                                                                                             .supplyAsync()
                                                                                                             .thenCombine(CompletableFuture
                                                                                                              .supplyAsync()
                                                                                                              .thenCombine(CompletableFuture
                                                                                                               .supplyAsync()
                                                                                                               .thenCombine(CompletableFuture
                                                                                                                .supplyAsync()
                                                                                                                .thenCombine(CompletableFuture
                                                                                                                 .supplyAsync()
                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                  .supplyAsync()
                                                                                                                  .thenCombine(CompletableFuture
                                                                                                                   .supplyAsync()
                                                                                                                   .thenCombine(CompletableFuture
                                                                                                                    .supplyAsync()
                                                                                                                    .thenCombine(CompletableFuture
                                                                                                                     .supplyAsync()
                                                                                                                     .thenCombine(CompletableFuture
                                                                                                                      .supplyAsync()
                                                                                                                      .thenCombine(CompletableFuture
                                                                                                                       .supplyAsync()
                                                                                                                       .thenCombine(CompletableFuture
                                                                                                                        .supplyAsync()
                                                                                                                        .thenCombine(CompletableFuture
                                                                                                                         .supplyAsync()
                                                                                                                         .thenCombine(CompletableFuture
                                                                                                                          .supplyAsync()
                                                                                                                          .thenCombine(CompletableFuture
                                                                                                                           .supplyAsync()
                                                                                                                           .thenCombine(CompletableFuture
                                                                                                                            .supplyAsync()
                                                                                                                            .thenCombine(CompletableFuture
                                                                                                                             .supplyAsync()
                                                                                                                             .thenCombine(CompletableFuture
                                                                                                                              .supplyAsync()
                                                                                                                              .thenCombine(CompletableFuture
                                                                                                                               .supplyAsync()
                                                                                                                               .thenCombine(CompletableFuture
                                                                                                                                .supplyAsync()
                                                                                                                                .thenCombine(CompletableFuture
                                                                                                                                 .supplyAsync()
                                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                                  .supplyAsync()
                                                                                                                                  .thenCombine(CompletableFuture
                                                                                                                                   .supplyAsync()
                                                                                                                                   .thenCombine(CompletableFuture
                                                                                                                                    .supplyAsync()
                                                                                                                                    .thenCombine(CompletableFuture
                                                                                                                                     .supplyAsync()
                                                                                                                                     .thenCombine(CompletableFuture
                                                                                                                                       .supplyAsync()
                                                                                                                                       .thenCombine(CompletableFuture
                                                                                                                                        .supplyAsync()
                                                                                                                                        .thenCombine(CompletableFuture
                                                                                                                                         .supplyAsync()
                                                                                                                                         .thenCombine(CompletableFuture
                                                                                                                                          .supplyAsync()
                                                                                                                                          .thenCombine(CompletableFuture
                                                                                                                                           .supplyAsync()
                                                                                                                                           .thenCombine(CompletableFuture
                                                                                                                                            .supplyAsync()
                                                                                                                                            .thenCombine(CompletableFuture
                                                                                                                                             .supplyAsync()
                                                                                                                                             .thenCombine(CompletableFuture
                                                                                                                                              .supplyAsync()
                                                                                                                                              .thenCombine(CompletableFuture
                                                                                                                                               .supplyAsync()
                                                                                                                                               .thenCombine(CompletableFuture
                                                                                                                                                .supplyAsync()
                                                                                                                                                .thenCombine(CompletableFuture
                                                                                                                                                 .supplyAsync()
                                                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                                                   .supplyAsync()
                                                                                                                                                   .thenCombine(CompletableFuture
                                                                                                                                                  .supplyAsync()
                                                                                                                                                  .thenCombine(CompletableFuture
                                                                                                                                                     .supplyAsync()
                                                                                                                                                     .thenCombine(CompletableFuture
                                                                                                                                                       .supplyAsync()
                                                                                                                                                       .thenCombine(CompletableFuture
                                                                                                                                                        .supplyAsync()
                                                                                                                                                        .thenCombine(CompletableFuture
                                                                                                                                                           .supplyAsync()
                                                                                                                                                           .thenCombine(CompletableFuture
                                                                                                                                                             .supplyAsync()
                                                                                                                                                             .thenCombine(CompletableFuture
                                                                                                                                                              .supplyAsync()
                                                                                                                                                              .thenCombine(CompletableFuture
                                                                                                                                                               .supplyAsync()
                                                                                                                                                               .thenCombine(CompletableFuture
                                                                                                                                                                 .supplyAsync()
                                                                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                                                                   .supplyAsync()
                                                                                                                                                                   .thenCombine(CompletableFuture
                                                                                                                                                                     .supplyAsync()
                                                                                                                                                                     .thenCombine(CompletableFuture
                                                                                                                                                                       .supplyAsync()
                                                                                                                                                                       .thenCombine(CompletableFuture
                                                                                                                                                                         .supplyAsync()
                                                                                                                                                                         .thenCombine(CompletableFuture
                                                                                                                                                                          .supplyAsync()
                                                                                                                                                                          .thenCombine(CompletableFuture
                                                                                                                                                                           .supplyAsync()
                                                                                                                                                                           .thenCombine(CompletableFuture
                                                                                                                                                                            .supplyAsync()
                                                                                                                                                                            .thenCombine(CompletableFuture
                                                                                                                                                                             .supplyAsync()
                                                                                                                                                                             .thenCombine(CompletableFuture
                                                                                                                                                                              .supplyAsync()
                                                                                                                                                                              .thenCombine(CompletableFuture
                                                                                                                                                                               .supplyAsync()
                                                                                                                                                                               .thenCombine(CompletableFuture
                                                                                                                                                                                .supplyAsync()
                                                                                                                                                                                .thenCombine(CompletableFuture
                                                                                                                                                                                 .supplyAsync()
                                                                                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                                                                                  .supplyAsync()
                                                                                                                                                                                  .thenCombine(CompletableFuture
                                                                                                                                                                                   .supplyAsync()
                                                                                                                                                                                   .thenCombine(CompletableFuture
                                                                                                                                                                                    .supplyAsync()
                                                                                                                                                                                    .thenCombine(CompletableFuture
                                                                                                                                                                                     .supplyAsync()
                                                                                                                                                                                     .thenCombine(CompletableFuture
                                                                                                                                                                                      .supplyAsync()
                                                                                                                                                                                      .thenCombine(CompletableFuture
                                                                                                                                                                                       .supplyAsync()
                                                                                                                                                                                       .thenCombine(CompletableFuture
                                                                                                                                                                                        .supplyAsync()
                                                                                                                                                                                        .thenCombine(CompletableFuture
                                                                                                                                                                                         .supplyAsync()
                                                                                                                                                                                         .thenCombine(CompletableFuture
                                                                                                                                                                                          .supplyAsync()
                                                                                                                                                                                          .thenCombine(CompletableFuture
                                                                                                                                                                                           .supplyAsync()
                                                                                                                                                                                           .thenCombine(CompletableFuture
                                                                                                                                                                                            .supplyAsync()
                                                                                                                                                                                            .thenCombine(CompletableFuture
                                                                                                                                                                                             .supplyAsync()
                                                                                                                                                                                             .thenCombine(CompletableFuture
                                                                                                                                                                                              .supplyAsync()
                                                                                                                                                                                              .thenCombine(CompletableFuture
                                                                                                                                                                                               .supplyAsync()
                                                                                                                                                                                               .thenCombine(CompletableFuture
                                                                                                                                                                                                .supplyAsync()
                                                                                                                                                                                                .thenCombine(CompletableFuture
                                                                                                                                                                                                 .supplyAsync()
                                                                                                                                                                                                 .thenCombine(CompletableFuture
                                                                                                                                                                                                       .supplyAsync()
                                                                                                                                       .thenCombine(CompletableFuture
                                                                          .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
                                                                  this::convert)
                          .orTimeout(3, SECONDS)
                          .whenComplete((amount, ex) -> {
                            if (amount != null)
                            { System.out.println("The price is: " + amount + "GBP"); }
                            else { System.out.println(ex.toString()); }
                          })
...
CompletableFuture
  .supplyAsync()
    () -> findBestPrice("LDN - NYC")
  .thenCombine(CompletableFuture
    .supplyAsync
      (() -> queryExchangeRateFor("GBP"))
  .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
        this::convert)
.orTimeout(3, SECONDS)
.whenComplete((amount, ex) -> {
  if (amount != null)
    { System.out.println("The price is: " + amount + "GBP"); } else
    { System.out.println(ex.toString()); } }
})

• This example asynchronously performs web services within a bounded time

  This method is always called, w/ or w/out an exception being thrown
Applying the orTimeout() & completeOnTimeout() Methods

- This example asynchronously performs web services within a bounded time

```java
CompletableFuture.supplyAsync()
    .thenCombine(CompletableFuture.supplyAsync()
                   .thenApply(result -> queryExchangeRateFor("GBP"))
                   .completeOnTimeout(DEFAULT_RATE, 2, SECONDS),
                   this::convert)
    .orTimeout(3, SECONDS)
    .whenComplete((amount, ex) -> {
        if (amount != null)
            { System.out.println("The price is: " + amount + "GBP"); } 
        else { System.out.println(ex.toString()); } 
    })...
```

Print results if async calls finished normally
CompletableFuture.supplyAsync()
  .thenCombine(CompletableFuture.supplyAsync()
    .thenApply(findBestPrice("LDN - NYC"))
    .thenApply(queryExchangeRateFor("GBP"))
    .whenComplete((amount, ex) -> {
      if (amount != null)
        System.out.println("The price is: " + amount + " GBP");
      else System.out.println(ex.toString());
    })))

• This example asynchronously performs web services within a bounded time

Print the exception if async call timed out
Applying completeAsync() to Big Fractions
### Applying completeAsync() to Big Fractions

- Using `completeAsync()` to multiply big fractions

```java
testFractionMultiplicationCompleteAsync() {
  StringBuilder sb = new StringBuilder(">> Starting test\n");
  String f1 = "62675744/15668936"; String f2 = "609136/913704";
  CompletableFuture<BigFraction> f = new CompletableFuture<>();
  f.thenRun(() -> sb.append("completeAsync() result = "));
  f.completeAsync(() -> {
    BigFraction bf1 = new BigFraction(f1);
    BigFraction bf2 = new BigFraction(f2);
    return bf1.multiply(bf2); });
  ...
  sb.append(f.join().toMixedString()); display(sb.toString());
}
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/Java8/ex8](https://github.com/douglascraigschmidt/LiveLessons/tree/master/Java8/ex8)
Applying completeAsync() to Big Fractions

- Using completeAsync() to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); }

    ... sb.append(f.join().toMixedString()); display(sb.toString());
}
```

This string builder holds intermediate results
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";
    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2);
    });
    ...
    sb.append(f.join().toMixedString()); display(sb.toString());
}
Applying completeAsync() to Big Fractions

- Using completeAsync() to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder("\nStarting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });

    sb.append(f.join().toMixedString()); display(sb.toString());
}
```

Create a new empty completable future
Using `completeAsync()` to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";
    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = ");
    f.completeAsync() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });
    ...
    sb.append(f.join().toMixedString()); display(sb.toString());
}
```

Register an action that appends a string when run (in calling thread)

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html#thenRun](https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html#thenRun)
Applying `completeAsync()` to Big Fractions

- Using `completeAsync()` to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder("\nStarting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";
    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });
    ...
    sb.append(f.join().toMixedString()); display(sb.toString());
}
```

> Arrange to execute a supplier lambda in common fork-join pool

See [docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#completeAsync](https://docs.oracle.com/javase/9/docs/api/java/util/concurrent/CompletableFuture.html#completeAsync)
Using `completeAsync()` to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });

    sb.append(f.join().toMixedString()); display(sb.toString());
}
```

This method sets all the processing in motion
Applying `completeAsync()` to Big Fractions

- Using `completeAsync()` to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });

    ... sb.append(f.join().toMixedString()); display(sb.toString());
}
```

This supplier lambda is used to multiply two BigFraction objects.
Applying completeAsync() to Big Fractions

- Using completeAsync() to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test
\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
f.thenRun(() -> sb.append("completeAsync() result = "));
f.completeAsync(() -> {
    BigFraction bf1 = new BigFraction(f1);
    BigFraction bf2 = new BigFraction(f2);
    return bf1.multiply(bf2); });

    ...sb.append(f.join().toMixedString()); display(sb.toString());
}
```

These computations run concurrently
Applying completeAsync() to Big Fractions

- Using `completeAsync()` to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync(() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); });

    ... join() blocks until result is complete
    sb.append(f.join().toMixedString()); display(sb.toString());
}
```
void testFractionMultiplicationCompleteAsync() { 
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";
    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); }
    ...
    sb.append(f.join().toMixedString()); display(sb.toString()); }

• Using completeAsync() to multiply big fractions
Applying completeAsync() to Big Fractions

• Using completeAsync() to multiply big fractions

```java
void testFractionMultiplicationCompleteAsync() {
    StringBuilder sb = new StringBuilder(">> Starting test\n");
    String f1 = "62675744/15668936"; String f2 = "609136/913704";

    CompletableFuture<BigFraction> f = new CompletableFuture<>();
    f.thenRun(() -> sb.append("completeAsync() result = "));
    f.completeAsync() -> {
        BigFraction bf1 = new BigFraction(f1);
        BigFraction bf2 = new BigFraction(f2);
        return bf1.multiply(bf2); }
    );
    ...
    sb.append(f.join().toMixedString()); display(sb.toString());
}
```

- Display output as a string
End of Understand
Enhancements to the Java
CompletableFuture Framework