

Table 6.1 Adding an action to a `CompletableFuture<T>` object

Method	Parameter	Description
<code>thenRun</code>	<code>() -> void</code>	After this Future completes, then execute the Runnable.
<code>thenAccept</code>	<code>T -> void</code>	Like <code>thenApply</code> , but with void result.
<code>thenApply</code>	<code>T -> U</code>	After this Future completes, then apply a function to its result.
<code>thenCompose</code>	<code>T -> CompletableFuture<U></code>	Invoke the Future valued function on the result and return its Future when it completes.
<code>whenComplete</code>	<code>(T, Throwable) -> void</code>	Like <code>handle</code> , but with void result.
<code>handle</code>	<code>(T, Throwable) -> U</code>	After this Future completes, process its result or error.
<code>exceptionally</code>	<code>Throwable -> T</code>	After this Future completes, process its error.

Table 6.2 Combining a `CompletableFuture<T>` object with a `CompletableFuture<U>` object

Method	Parameters	Description
<code>runAfterBoth</code>	<code>CompletableFuture<?>, () -> void</code>	After both Futures complete, then execute the Runnable.
<code>thenAcceptBoth</code>	<code>CompletableFuture<U>, (T, U) -> void</code>	Like <code>thenCombine</code> , but with void result.
<code>thenCombine</code>	<code>CompletableFuture<U>, (T, U) -> V</code>	After both Futures complete, combine the results with the given function.
<code>runAfterEither</code>	<code>CompletableFuture<?>, () -> void</code>	After one or the other Future completes, then execute the Runnable.
<code>acceptEither</code>	<code>CompletableFuture<T>, T -> void</code>	Like <code>applyToEither</code> , but with void result.
<code>applyToEither</code>	<code>CompletableFuture<T>, T -> V</code>	After one or the other Future completes, then apply a function to its result.
<code>static allOf</code>	<code>CompletableFuture<?>...</code>	Complete with void result after all given Futures complete.
<code>static anyOf</code>	<code>CompletableFuture<?>...</code>	Complete with void result after any given Future completes.