
T
<<interface>>
CompletionStage
+ thenApply(fn : Function<T,U>) : CompletionStage<U>
+ thenApplyAsync(fn : Function<T,U>) : CompletionStage<U>
+ thenApplyAsync(fn : Function<T,U>, executor : Executor) : CompletionStage<U>
+ thenAccept(action : Consumer<T>) : CompletionStage<Void>
+ thenAcceptAsync(action : Consumer<T>) : CompletionStage<Void>
+ thenAcceptAsync(action : Consumer<T>, executor : Executor) : CompletionStage<Void>
+ thenRun(action : Runnable) : CompletionStage<Void>
+ thenRunAsync(action : Runnable) : CompletionStage<Void>
+ thenRunAsync(action : Runnable, executor : Executor) : CompletionStage<Void>
+ thenCombine(other : CompletionStage<U>, fn : BiFunction<T,U,V>) : CompletionStage<V>
+ thenCombineAsync(other : CompletionStage<U>, fn : BiFunction<T,U,V>) : CompletionStage<V>
+ thenCombineAsync(other : CompletionStage<U>, fn : BiFunction<T,U,V>, executor : Executor) : CompletionStage<V>
+ thenAcceptBoth(other : CompletionStage<U>, action : BiConsumer<T,U>) : CompletionStage<Void>
+ thenAcceptBothAsync(other : CompletionStage<U>, action : BiConsumer<T,U>) : CompletionStage<Void>
+ thenAcceptBothAsync(other : CompletionStage<U>, action : BiConsumer<T,U>, executor : Executor) : CompletionStage<Void>
+ runAfterBoth(other : CompletionStage<?>, action : Runnable) : CompletionStage<Void>
+ runAfterBothAsync(other : CompletionStage<?>, action : Runnable) : CompletionStage<Void>
+ runAfterBothAsync(other : CompletionStage<?>, action : Runnable, executor : Executor) : CompletionStage<Void>
+ applyToEither(other : CompletionStage<T>, fn : Function<T,U>) : CompletionStage<U>
+ applyToEitherAsync(other : CompletionStage<T>, fn : Function<T,U>) : CompletionStage<U>
+ applyToEitherAsync(other : CompletionStage<T>, fn : Function<T,U>, executor : Executor) : CompletionStage<U>
+ acceptEither(other : CompletionStage<T>, action : Consumer<T>) : CompletionStage<Void>
+ acceptEitherAsync(other : CompletionStage<T>, action : Consumer<T>) : CompletionStage<Void>
+ acceptEitherAsync(other : CompletionStage<T>, action : Consumer<T>, executor : Executor) : CompletionStage<Void>
+ runAfterEither(other : CompletionStage<?>, action : Runnable) : CompletionStage<Void>
+ runAfterEitherAsync(other : CompletionStage<?>, action : Runnable) : CompletionStage<Void>
+ runAfterEitherAsync(other : CompletionStage<?>, action : Runnable, executor : Executor) : CompletionStage<Void>
+ thenCompose(fn : Function<T,CompletionStage<U>>) : CompletionStage<U>
+ thenComposeAsync(fn : Function<T,CompletionStage<U>>) : CompletionStage<U>
+ thenComposeAsync(fn : Function<T,CompletionStage<U>>, executor : Executor) : CompletionStage<U>
+ handle(fn : BiFunction<T,Throwable,U>) : CompletionStage<U>
+ handleAsync(fn : BiFunction<T,Throwable,U>) : CompletionStage<U>
+ handleAsync(fn : BiFunction<T,Throwable,U>, executor : Executor) : CompletionStage<U>
+ whenComplete(action : BiConsumer<T,Throwable>) : CompletionStage<T>
+ whenCompleteAsync(action : BiConsumer<T,Throwable>) : CompletionStage<T>
+ whenCompleteAsync(action : BiConsumer<T,Throwable>, executor : Executor) : CompletionStage<T>
+ exceptionally(fn : Function<Throwable,T>) : CompletionStage<T>
+ toCompletableFuture() : CompletableFuture<T>