

# COMPLETABLEFUTURE API IMPROVEMENTS

[http://www.tutorialspoint.com/java9/java9\\_completablefuture\\_api\\_improvements.htm](http://www.tutorialspoint.com/java9/java9_completablefuture_api_improvements.htm)

Copyright © tutorialspoint.com

## Advertisements

CompletableFuture class was introduced in Java 8 to represent the Future which can be completed by setting its value and status explicitly. It can be used as `java.util.concurrent.CompletionStage`. It supports dependent functions and actions which got triggered upon the future's completion. In java 9 CompletableFuture API has been enhanced further. Following are the relevant changes done to the API.

- Support for delays and timeouts.
- Improved support for subclassing.
- New factory methods added.

## Support for delays and timeouts

```
public CompletableFuture<T> completeOnTimeout(T value, long timeout, TimeUnit unit)
```

This method completes this `CompletableFuture` with the given value if not otherwise completed before the given timeout.

```
public CompletableFuture<T> orTimeout(long timeout, TimeUnit unit)
```

This method exceptionally completes this `CompletableFuture` with a `TimeoutException` if not otherwise completed before the given timeout.

## Improved support for subclassing

```
public Executor defaultExecutor()
```

It returns the default `Executor` used for async methods that do not specify an `Executor`. This method may be overridden in subclasses to return an `Executor` to provide one independent thread as minimum.

```
public <U> CompletableFuture<U> newIncompleteFuture()
```

Returns a new incomplete `CompletableFuture` of the type to be returned by a `CompletionStage` method. Subclasses of `CompletableFuture` class should override this method to return an instance of the same class as this `CompletableFuture`. The default implementation returns an instance of class `CompletableFuture`.

## New factory Methods

```
public static <U> CompletableFuture<U> completedFuture(U value)
```

This factory method returns a new `CompletableFuture` which is already completed with the given value.

```
public static <U> CompletionStage<U> completedStage(U value)
```

This factory method returns a new `CompletionStage` which is already completed with the given value and supports only those methods present in interface `CompletionStage`.

```
public static <U> CompletionStage<U> failedStage(Throwable ex)
```

This factory method returns a new `CompletionStage` which is already completed exceptionally with the given exception and supports only those methods present in interface `CompletionStage`.