# Java 9 改进的 CompletableFuture API

Java 8 引入了 **CompletableFuture<T>** 类，可能是 **java.util.concurrent.Future<T>** 明确的完成版（设置了它的值和状态），也可能被用作**java.util.concurrent.CompleteStage** 。支持 future 完成时触发一些依赖的函数和动作。Java 9 引入了一些**CompletableFuture** 的改进：

Java 9 对 **CompletableFuture** 做了改进：

* 支持 delays 和 timeouts
* 提升了对子类化的支持
* 新的工厂方法

### **支持 delays 和 timeouts**

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

在 **timeout**（单位在 **java.util.concurrent.Timeunits units** 中，比如 MILLISECONDS ）前以给定的 value 完成这个 CompletableFutrue。返回这个 CompletableFutrue。

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

如果没有在给定的 timeout 内完成，就以 java.util.concurrent.TimeoutException 完成这个 CompletableFutrue，并返回这个 CompletableFutrue。

### **增强了对子类化的支持**

做了许多改进使得 **CompletableFuture** 可以被更简单的继承。比如，你也许想重写新的 **public Executor defaultExecutor()** 方法来代替默认的 **executor**。

另一个新的使子类化更容易的方法是：

public <U> CompletableFuture<U> newIncompleteFuture()

### **新的工厂方法**

Java 8引入了 **<U> CompletableFuture<U> completedFuture(U value)** 工厂方法来返回一个已经以给定 value 完成了的 **CompletableFuture**。Java 9以 一个新的 **<U> CompletableFuture<U> failedFuture(Throwable ex)** 来补充了这个方法，可以返回一个以给定异常完成的 **CompletableFuture**。

除此以外，Java 9 引入了下面这对 stage-oriented 工厂方法，返回完成的或异常完成的 completion stages:

* **<U> CompletionStage<U> completedStage(U value)**: 返回一个新的以指定 value 完成的**CompletionStage** ，并且只支持 **CompletionStage** 里的接口。
* **<U> CompletionStage<U> failedStage(Throwable ex)**: 返回一个新的以指定异常完成的**CompletionStage** ，并且只支持 **CompletionStage** 里的接口。