

CompletableFuture

supplyAsync

有返回值

runAsync

不支持返回值

completableFuture获取结果

join：抛出的是unchecked异常（即未经检查的异常），不会强制开发者抛出

get：抛出的是受查异常，
ExecutionException, InterruptedException 需
要用户手动处理（抛出或者 try catch）

计算结果完成后的处理方法

whenComplete

当completableFuture的计算结果完成，或者抛
异常时，可以执行特定的action

whenCompleteAsync

把whenCompleteAsync的任务action继续提交
给线程池来执行，即执行CompletableFuture的
任务和执行whenCompleteAsync的任务的线程
可能是两个不同的线程

thenApply

当线程B依赖于线程A的执行结果时，可以使用
thenApply方法来把这两个线程串行化

thenApplyAsync

线程异步

handle

handle方法和thenApply方法处理方式基本一
样，不同的是handle里的方法是在
supplyAsync/runAsync执行后一定会执行的，
即使supplyAsync/runAsync里抛了异常也会执
行handle里的方法，而thenApply只可以执行正
常的任务，任务出现异常则不执行thenApply方
法

handleAsync

处理线程异步

thenAccept

thenAccept方法同样也是对前面的
supplyAsync/runAsync生成的结果进行消费，
但是不同点在于thenAccept方法只是纯消费，
不返回值

thenCompose

thenCompose()用来连接两个
CompletableFuture，返回值是新的
CompletableFuture；

thenApply()转换的是泛型中的类型，是同一个
CompletableFuture。

thenCombine

thenCombine会把两个CompletableFuture的
任务都执行完成后，把两个任务的结果一块交给
thenCombine来处理，并生成新的
CompletableFuture任务。

allOf

allOf方法的入参是若干个CompletableFuture
任务，返回类型是CompletableFuture<
Void>，allOf方法是等所有的
CompletableFuture都执行完后再执行计算，一
般后面会跟链式的thenApply方法或者
thenAccept方法对所有的异步任务进行汇总处
理

anyOf

anyOf方法入参是若干个CompletableFuture任
务，返回类型是CompletableFuture<
Object>，anyOf方法只要有一个
CompletableFuture任务完后就执行计算，一般
后面会跟链式的thenApply方法或者thenAccept
方法对结果进行处理，anyOf方法没allOf方法使
用广泛。