Future vs CompletableFuture

CompletableFuture is an extension to Java’s Future API, The Future is used as a reference to the result of an asynchronous computation. It provides isDone() to check computation is done or not and get() to retrieve the result. But the limitations of Future which are:

1.It cannot be manually completed.

2.You cannot perform further action on a Future’s result without blocking

3. Multiple Futures cannot be chained together.

4.You can not combine multiple Futures together.

5. No Exception Handling.

P.S. Asynchronous programming is a means of writing non-blocking code by running a task on a separate thread than the main application thread and notifying the main thread about its progress, completion or failure.

CompletableFuture implements Future and CompletionStage interfaces and provides a huge set of convenience methods for creating, chaining and combining multiple Futures. It also has a very comprehensive exception handling support.

For example:

runAsync() - Running asynchronous computation.

supplyAsync() - Run a task asynchronously and return the result.

thenApply() - process and transform the result of a CompletableFuture when it arrives.

thenAccept(), thenRun() - you don’t want to return anything from your callback function and just want to run some piece of code after the completion of the Future.