

Exercise 1 : Implementing map and filter on Futures

In this exercise, you will come up with an implementation of the map and filter methods of Futures. First of all, spend some time as a group to make sure that you understand what those methods are supposed to do. Then, complete the following code to implement the two methods:

```
trait Future[T] { self =>
  def map[S](f: T => S): Future[S] =
    new Future[S] {
      def onComplete(callback: Try[S] => Unit): Unit = ???
    }

  def filter(f: T => Boolean): Future[T] =
    new Future[T] {
      def onComplete(callback: Try[T] => Unit): Unit = ???
    }
}
```

Solution

```
trait Future[T] { self =>
  def map[S](f: T => S): Future[S] =
    new Future[S] {
      def onComplete(callback: Try[S] => Unit): Unit = self.onComplete {
        case Success(v) => callback(Success(f(v)))
        case Failure(e) => callback(Failure(e))
      }
    }

  def filter(f: T => Boolean): Future[T] =
    new Future[T] {
      def onComplete(callback: Try[T] => Unit): Unit = self.onComplete {
        case Success(v) =>
          if f(v) {
            callback(Success(v))
          }
          else {
            callback(Failure(new NoSuchElementException("...")))
          }
        case Failure(e) => callback(Failure(e))
      }
    }
}
```