**CSYE 7200: Big Data System Engineering Using Scala**

**Section 1**

**Assignment 6**

Rajendra Kumar Rajkumar

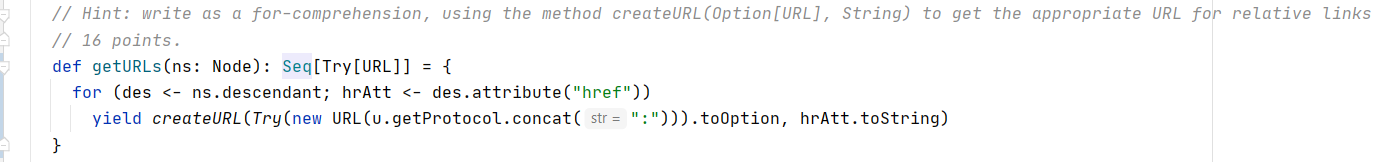
001405755

# **Implementation for method ‘getURLs()’ in WebCrawler.scala (line: 28):**

## **Code Snippet:**

*// Hint: write as a for-comprehension, using the method createURL(Option[URL], String) to get the appropriate URL for relative links  
// 16 points.*def getURLs(ns: Node): Seq[Try[URL]] = {  
 for (des <- ns.descendant; hrAtt <- des.attribute("href"))  
 yield *createURL*(*Try*(new URL(u.getProtocol.concat(":"))).toOption, hrAtt.toString)  
}

## **Screenshot of Code Snippet from IntelliJ:**

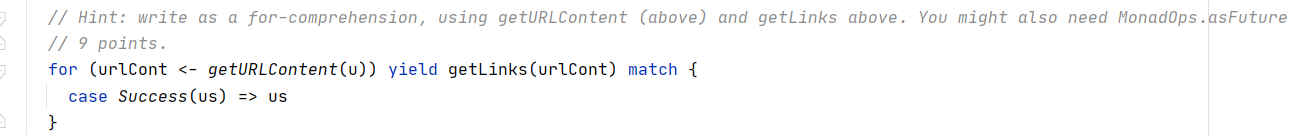


# **Implementation for method ‘wget()’ in WebCrawler.scala (line: 39):**

## **Code Snippet:**

*// Hint: write as a for-comprehension, using getURLContent (above) and getLinks above. You might also need MonadOps.asFuture  
// 9 points.*for (urlCont <- *getURLContent*(u)) yield getLinks(urlCont) match {  
 case *Success*(us) => us  
}

## **Screenshot of Code Snippet from IntelliJ:**

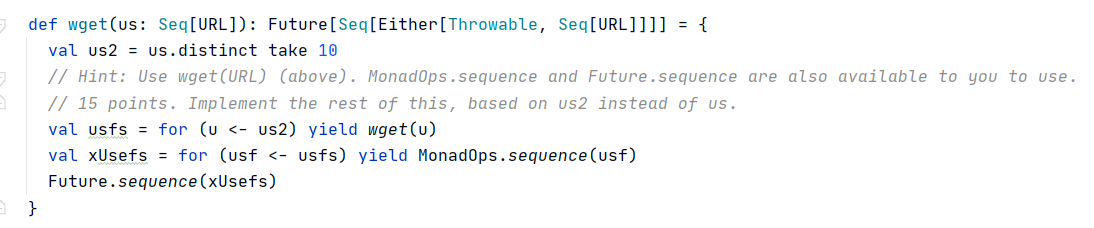


# **Implementation for method ‘wget()’ in WebCrawler.scala (line: 44):**

## **Code Snippet:**

def wget(us: Seq[URL]): Future[Seq[Either[Throwable, Seq[URL]]]] = {  
 val us2 = us.distinct take 10  
 *// Hint: Use wget(URL) (above). MonadOps.sequence and Future.sequence are also available to you to use.  
 // 15 points. Implement the rest of this, based on us2 instead of us.* val usfs = for (u <- us2) yield *wget*(u)  
 val xUsefs = for (usf <- usfs) yield MonadOps.*sequence*(usf)  
 Future.*sequence*(xUsefs)  
}

## **Screenshot of Code Snippet from IntelliJ:**

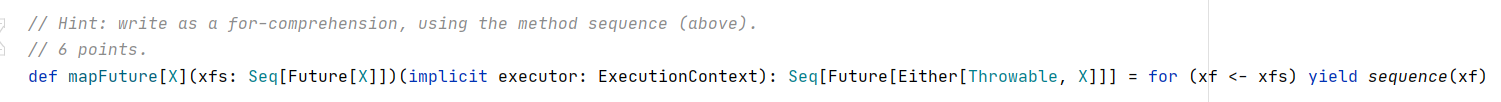


# **Implementation for method ‘mapFuture()’ in MonadOps.scala (line: 56):**

## **Code Snippet:**

*// Hint: write as a for-comprehension, using the method sequence (above).  
// 6 points.*def mapFuture[X](xfs: Seq[Future[X]])(implicit executor: ExecutionContext): Seq[Future[Either[Throwable, X]]] = for (xf <- xfs) yield *sequence*(xf)

## **Screenshot of Code Snippet from IntelliJ:**

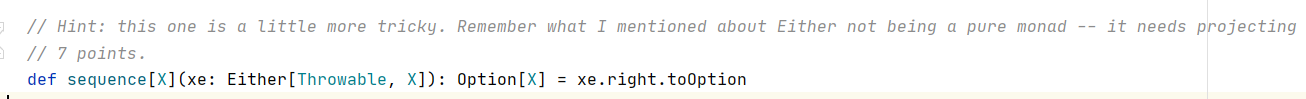


# **Implementation for method ‘sequence()’ in MonadOps.scala (line: 91):**

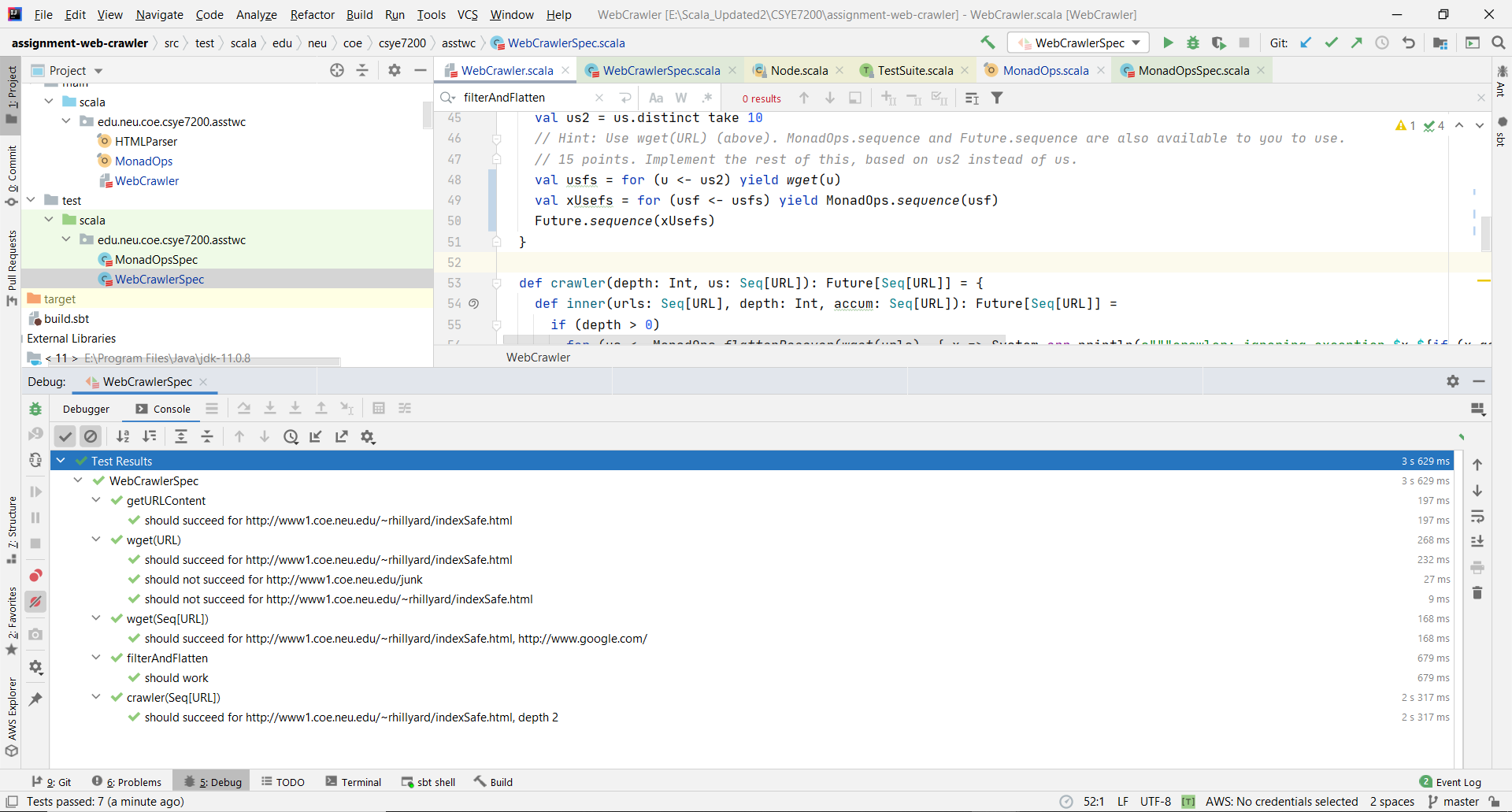
## **Code Snippet:**

*// Hint: this one is a little more tricky. Remember what I mentioned about Either not being a pure monad -- it needs projecting  
// 7 points.*def sequence[X](xe: Either[Throwable, X]): Option[X] = xe.right.toOption

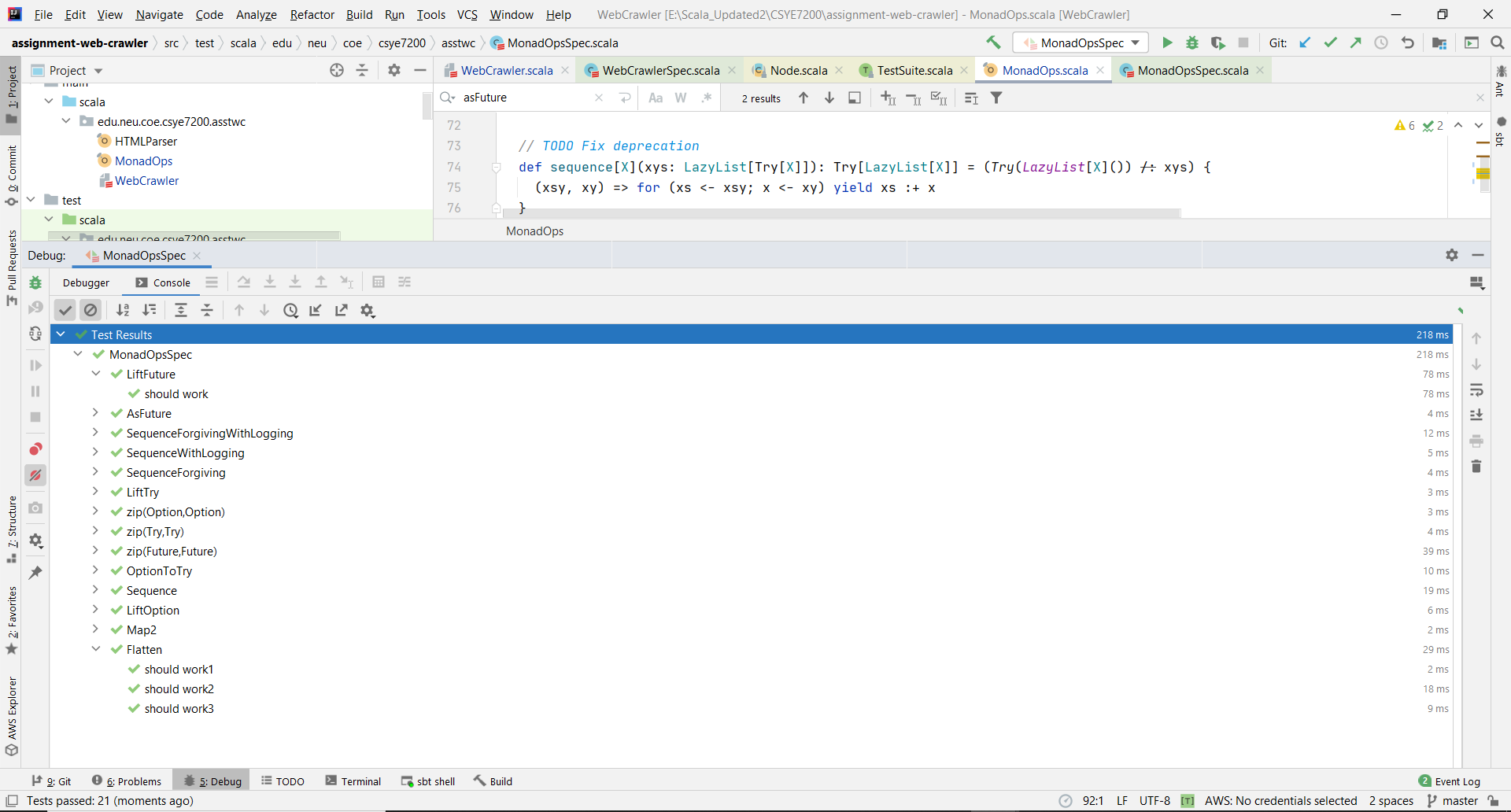
## **Screenshot of Code Snippet from IntelliJ:**



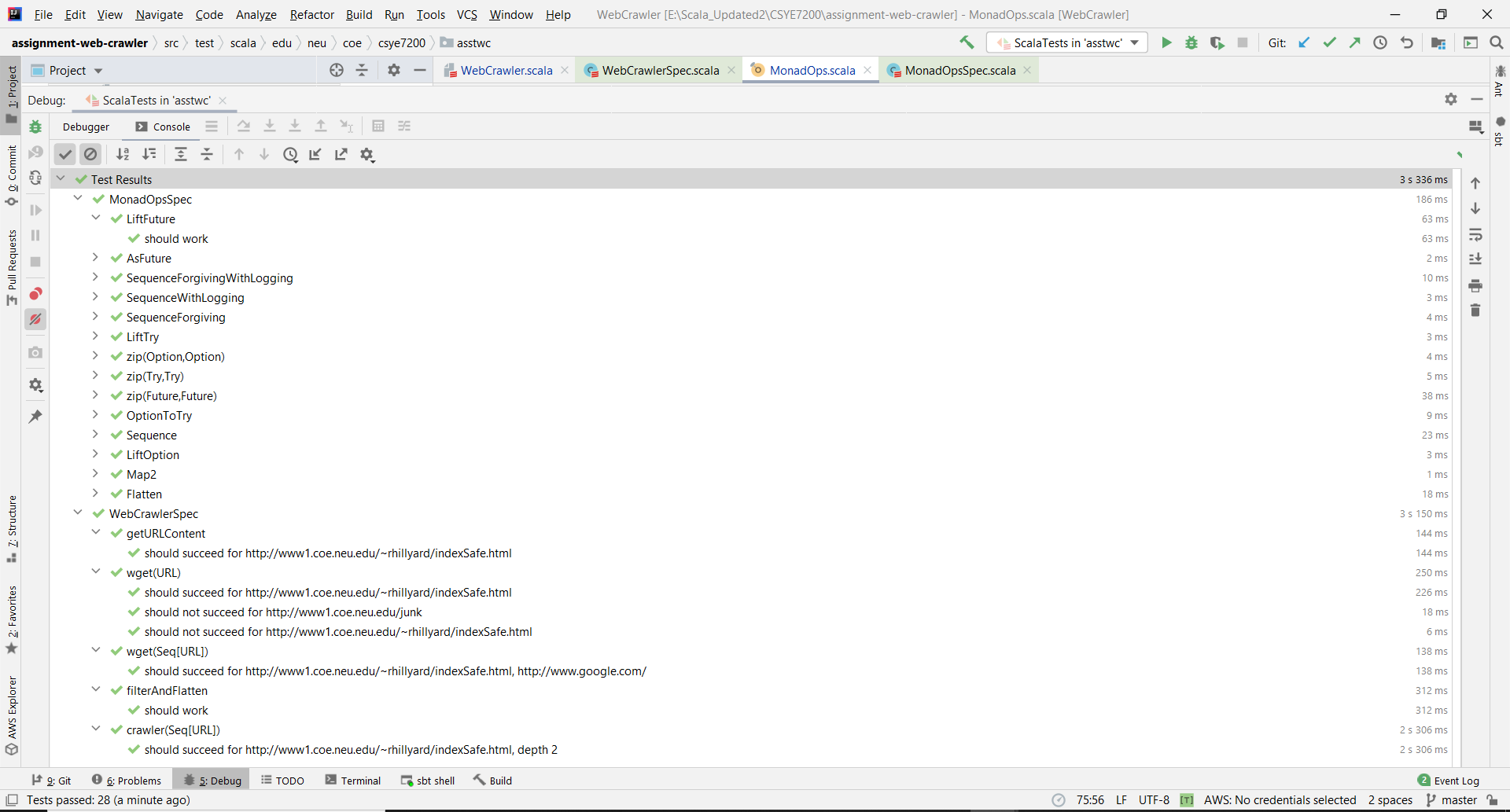
# **Execution of unit test class ‘WebCrawlerSpec.scala’ denoting that all 7 tests have passed:**



# **Execution of unit test class ‘MonadOpsSpec.scala’ denoting that all 21 tests have passed:**



# **Execution of unit test package ‘asstwc’ denoting that all 28 tests from MonadOpsSpec.Scala and WebCrawlerSpec have passed:**



# **References:**

1. <https://stackoverflow.com/questions/20874186/scala-listfuture-to-futurelist-disregarding-failed-futures>