Sai Sumana Puppala – SEC01 (NUID 002925158)

Big Data System Engineering with Scala  
Fall 2022   
Assignment No. 6



**-List of Tasks Implemented**

Implemented the 3 code changes.

def getURLs(ns: Node): Seq[Try[URL]] = for (n <- ns \\ "a"; nl = n \@ "href") yield *validateURL*(*createURL*(Some(url), nl.toString()))

for(l <- *getURLContent*(url); ls <- MonadOps.*asFuture*(getLinks(l))) yield ls// TO BE IMPLEMENTED  
}

def asOption[X](xe: Either[Throwable, X]): Option[X] = xe.toOption// TO BE IMPLEMENTED

Github Link:

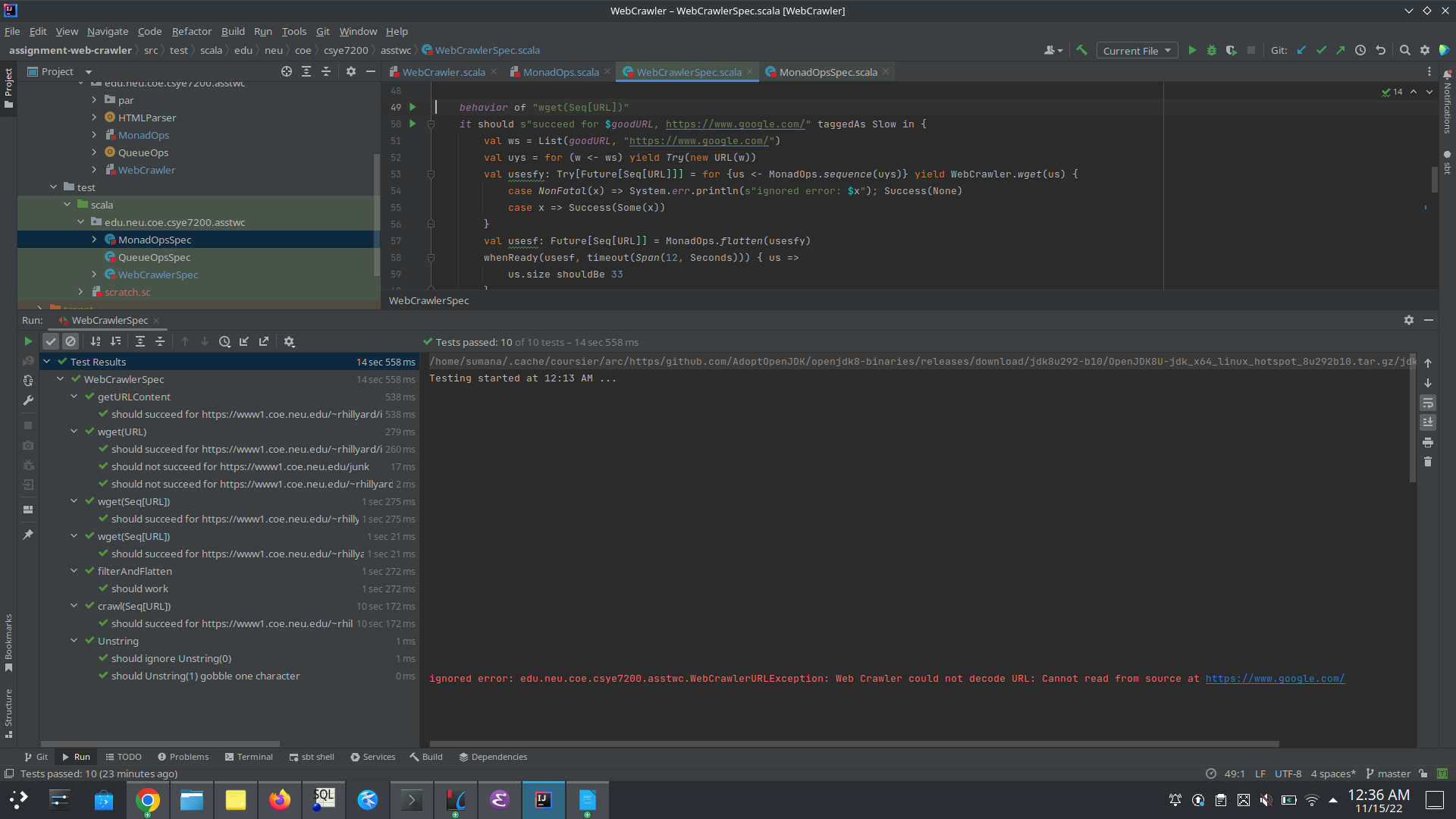
https://github.com/puppala-sumana/CSYE7200

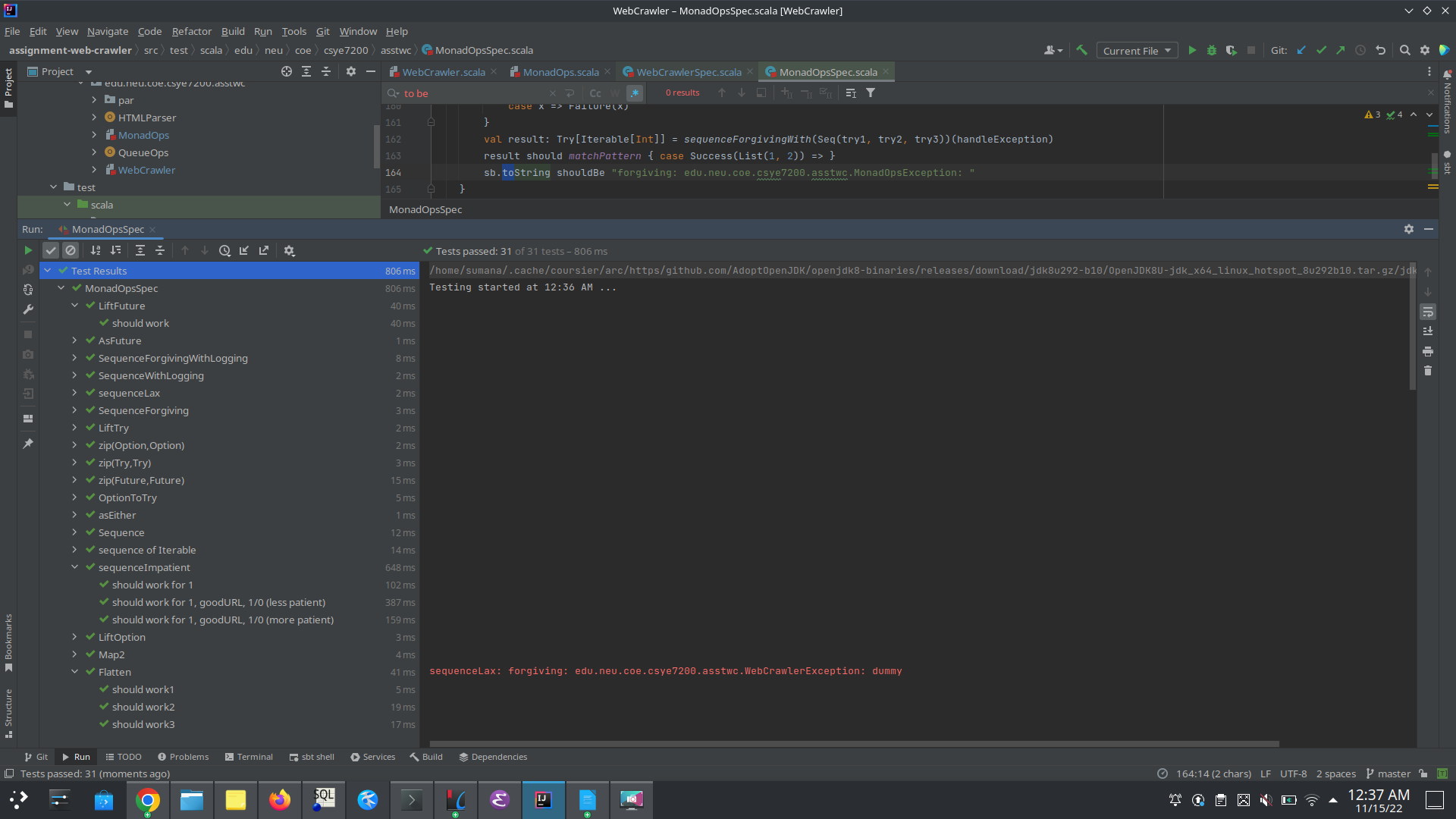
**-Code**

def wget(url: URL)(implicit ec: ExecutionContext): Future[Seq[URL]] = {  
 // 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 (n <- ns \\ "a"; nl = n \@ "href") yield *validateURL*(*createURL*(Some(url), nl.toString())) // TO BE IMPLEMENTED  
  
 def getLinks(g: String): Try[Seq[URL]] = {  
 val ny: Try[Node] = HTMLParser.*parse*(g) recoverWith { case f => Failure(new RuntimeException(s"parse problem with URL **$**url: **$**f")) }  
 for (n <- ny; uys = getURLs(n); us <- MonadOps.*sequenceForgiveSubsequent*(uys) { case \_: WebCrawlerProtocolException => true; case \_ => false }) yield us  
 }  
 // Hint: write as a for-comprehension, using getURLContent (above) and getLinks above. You will also need MonadOps.asFuture  
 // 9 points.  
 for(l <- *getURLContent*(url); ls <- MonadOps.*asFuture*(getLinks(l))) yield ls// TO BE IMPLEMENTED  
}

def asOption[X](xe: Either[Throwable, X]): Option[X] = xe.toOption// TO BE IMPLEMENTED

**-Unit tests**



****

**- Crawler Suggestions:**

**wget(url: URL)(implicit ec: ExecutionContext): Future[Seq[URL]] can be parallelized to fetch reference links individually through multiple crawler processes. This can be implemented through Spark, using RDD partitions. Each partition can crawl the reference link to check for urlContent and execute the respective Futures. Each future would run parallelly and we can waitUntil all the futures are executed and get the respective results. This way, wget would make each network call parallelly.**