Examples With Future

Effective Programming in Scala

Fetching Paginated Resources

You want to fetch data from a web service.

The web service returns data in pages, and provides the following method to fetch one specific page:

```
def getPage(page: Int): Future[String]
```

The service also provides the following method to know how many pages there are, in total:

```
def getPagesCount(): Future[Int]
```

Exercise: write a program that fetches all the pages.

```
def getAllPages(): Future[Seq[String]] = ???
```

Fetching Paginated Resources (Parallel Solution)

We first get the pages count, and then we fetch each page with `getPage`.

```
def getAllPages(): Future[Seq[String]] =
  getPagesCount().flatMap { pagesCount =>
    Future.traverse(1 to pagesCount)(getPage)
  }
```

Fetching Paginated Resources (Parallel Solution)

We first get the pages count, and then we fetch each page with `getPage`.

```
def getAllPages(): Future[Seq[String]] =
  getPagesCount().flatMap { pagesCount =>
    Future.traverse(1 to pagesCount)(getPage)
  }
```

What is the parallelism level?

This will be discussed in the next lesson.

Fetching Paginated Resources (Sequential Solution)

We first get the pages count, and then we build a long chain of computations fetching each page.

```
def getAllPages(): Future[Seq[String]] =
  getPagesCount().flatMap { pagesCount =>
    (1 to pagesCount).foldLeft(Future.successful(Seq.empty[String])) {
      (eventualPreviousPages, pageNumber) =>
        eventualPreviousPages.flatMap { previousPages =>
          getPage(pageNumber)
            .map(pageContent => previousPages :+ pageContent)
        }
    }
  }
```

Retry a Couple of Times

The web service is flaky. In the previous example, if fetching a page fails, the `getAllPages` method returns a failed `Future`.

Instead, you want to retry at most 3 times before giving up.

```
def resilientGetAllPages(): Future[Seq[String]] = ???
```

Retry a Couple of Times (Solution)

```
def resilientGetPage(page: Int): Future[String] =
  val maxAttempts = 3
  def attempt(remainingAttempts: Int): Future[String] =
    if remainingAttempts == 0 then
      Future.failed(Exception(s"Failed after $maxAttempts"))
    else
      println("Trying to fetch page $page")
      println(s"($remainingAttempts remaining attempts)")
      getPage(page).recoverWith { case NonFatal(_) =>
        System.err.println(s"Fetching page $page failed...")
        attempt(remainingAttempts - 1)
      }
  attempt(maxAttempts)
```