Programming Assignment: Programming with Futures (verified track)

Deadline Pass this assignment by Sep 20, 8:59 AM CEST

1.

- • The goal of this assignment is to familiarize yourself with the infrastructure and the tools required during this class.

Installing Required Tools

Before anything else, *it is crucial that you make sure that all tools are correctly installed.* Take a careful look at the Tools Setup pages and verify that all of the listed tools work on your machine.

The Assignment

Part 1: Obtain the Project Files

To get started, download the reactive-async.zip handout archive file and extract it somewhere on your machine.

Part 2: Editing the Project

In case you like using an IDE, import the project into your IDE. If you are new to Scala and don't know which IDE to choose, we recommend to try IntelliJ. You can find information on how to import a Scala project into IntelliJ in the IntelliJ IDEA Tutorial page.

Then, in the folder *src/main/scala*, open the package *async* and double-click the file *Async.scala*. This files contains an object whose methods need to be implemented.

When working on an assignment, it is important that you don't change any existing method, class or object names or types. When doing so, our automated grading tools will not be able to recognize your code and you have a high risk of not obtaining any points for your solution.

Part 3: Running your Code

Once you start writing some code, you might want to experiment with Scala, execute small snippets of code, or also run some methods that you already implemented. We present two possibilities to run Scala code.

Note that these tools are recommended for exploring Scala, but should not be used for testing your code. The next section of this document will explain how to write tests in Scala.

Using the Scala REPL

In the sbt console, start the Scala REPL by typing *console*:

> console [info] Starting scala interpreter... scala>

In the REPL you can try out arbitrary snippets of Scala code, for example:

```
scala> val l = List(3,7,2)
val l: List[Int] = List(3, 7, 2)
scala> l.isEmpty
val res0: Boolean = false
scala> l.tail.head
val res1: Int = 7
scala> List().isEmpty
```

`val res2: Boolean = true`The classes of the assignment are available inside the REPL, so you can for instance import all the methods from object *Async*:

```
scala> import async.Async.*
scala> transformSuccess(concurrent.Future.successful(3))
```

`val res1: scala.concurrent.Future[Boolean] = Future(Success(false))`In order to exit the Scala

REPL and go back to sbt, just type *ctrl-d*.

Using a Main Object

Another way to run your code is to create a new Main object that can be executed by the Java Virtual Machine.

In your code editor, create a new *Main.scala* file in the *async* package:

```
package async
@main def Main =
  println(Async.transformSuccess(concurrent.Future.successful(3)))
```

You can run the *Main* object in the sbt console by simply using the command *run*.

Part 5: Testing your Code

Throughout the assignments of this course we will require you to write unit tests for the code that you write. Unit tests are the preferred way to test your code because unlike REPL commands, unit tests are saved and can be re-executed as often as required. This is a great way to make sure that nothing breaks when you have go back later to change some code that you wrote earlier on.

We will be using the ScalaTest testing framework to write our unit tests. Navigate to the folder *src/test/scala* and open the file *AsyncSuite.scala* in package *async*. This file contains a some tests for the methods that need to be implemented.

You can run the tests by invoking the *test* sbt command. A test report will be displayed, showing you which tests failed and which tests succeeded.

Part 6: Submitting your Solution

Once you implemented all the required methods and tested you code thoroughly, you can submit it to our graders. Use sbt to submit your code as described in the "SBT tutorial and Submission of Assignments" section of the Tools Setup module.