CEN3024C Module 6 Project

For this project we will be implementing logging and profiling to the PizzaRestaurant project.

- Start by adding a few logging statements to the Customer Controller when a customer is created; add an INFO level statement for the success case and and ERROR level statement for the error case.

  Here are some tips to get you started:

  Exclude spring-boot-starter-logging in the pom file so Logback won't be used

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <exclusions>
        <exclusion>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-logging</artifactId>
        </exclusion>
    </exclusions>
</dependency>
```

  Add a dependency for Log4j2:

```
<!-- Add Log4j2 dependency -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-log4j2</artifactId>
</dependency>
```

  These are the import statements I added to my controller class:

```
// SLF4J Logger and LoggerFactory
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

  Declare a logger instance variable for the class:

```
 // Declare a Logger instance
 private static final Logger logger =
     LoggerFactory.getLogger(CustomerController.class);
```

  Sample logging statements look like this (change info to error for an error entry) (red text is mine for emphasis in this document, but you could use a pattern to add this to the log!):

```
logger.info("A new customer has been added: {}", createdCustomer.getUserId());

2024-03-12T15:57:17.893-04:00  INFO 10412 --- [nio-8080-exec-1]
e.f.c.m.controller.CustomerController    : A new customer has been added: john89

2024-03-12T16:18:25.083-04:00 ERROR 27908 --- [nio-8080-exec-1]
e.f.c.m.controller.CustomerController    : Error creating customer
edu.fscj.cen3024c.m6pizzarestaurant.exceptions.InvalidDataException: invalid data!
```

  In my controller I added test code to force exceptions to be thrown based on specific userId values:

```
if (customerDTO.getUserId().equals("testinvaliddata"))
        throw new InvalidDataException("invalid data!");
else if (customerDTO.getUserId().equals("testnosuchalgorithm"))
    throw new InvalidDataException("no such algorithm!");
```

- After adding the logging statements, add a profiler to the controller to track the time required to create a new customer, e.g.,

```
2024-03-12T16:58:07.629-04:00  INFO 22184 --- [nio-8080-exec-1]
e.f.c.m.controller.CustomerController    : A new customer has been added:
brucebrennan
+ Profiler [createCustomer]
|-- elapsed time        [Create Customer]    35.790 milliseconds.
|-- Total                [createCustomer]    36.151 milliseconds.
```

  These are the import statements I added to my controller class:

```
// SLF4J Profiler
import org.slf4j.profiler.Profiler;
import org.slf4j.profiler.TimeInstrument;
```

  At the top of the createCustomer method I start the profiler:

```
Profiler profiler = new Profiler("createCustomer");
profiler.start("Create Customer");
```

  Since a try statement is in this method, I added the necessary code to stop the profiler in a finally to ensure it was not skipped:

```
finally {
    TimeInstrument ti = profiler.stop();
    ti.print();
}
```

Submit your project and screensnips showing your test results to the GitHub classroom repo.