

|  |  |
| --- | --- |
| Employee Name: | MANOJ KUMAR YEKOLLU |
| Training Supervised by: | **KRISHNA BODDULURI** |
| Training Task #: | **ASSIGNMENT02052020** |
| Training Resource Materials: | **Spring Boot Caching:**  **https://spring.io/guides/gs/caching/ - Spring Boot with Cache**  **https://www.candidjava.com/tutorial/spring-boot-quartz-scheduler-example-code/ - Spring Boot with Quartz Scheduler** |
| Training Task Date: | **02/05/2020** |
| Task Due Date: | **02/15/2020** |
| Task Submitted Date: | **02/15/2020** |
| Github link: | [**https://github.com/NightFury546/Spring-Boot-Quartz.git**](https://github.com/NightFury546/Spring-Boot-Quartz.git)  [**https://github.com/NightFury546/gs-caching.git**](https://github.com/NightFury546/gs-caching.git) |
| Technologies used for Training | **Spring boot, maven, java, quartz** |

**Task Description/Requirement:**

**Develop an Application using Spring to demonstrate the caching process and spring boot Quartz scheduler.**

**High Level Synopsis:**

1. **Spring Boot Caching**

To enable caching in for spring boot application, spring boot provided @EnableCaching annotation.

This enable caching annotation @EnableCaching triggers a pre-processor that inspects every spring bean for caching annotation on public methods.

If such an annotation found, a proxy is automatically create to intercept the method call and handle the caching behavior accordingly.

The Caching annotation that the pre-processor look for are

@Cacheable

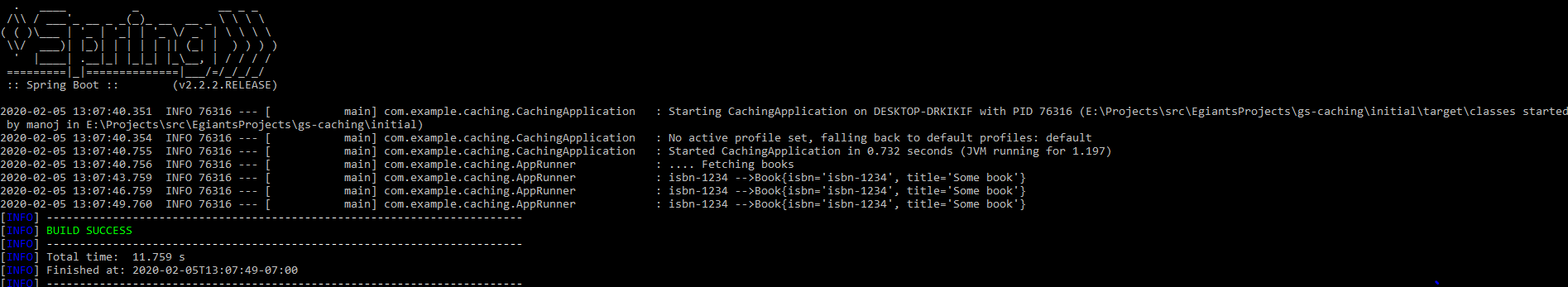
@CachePut

@CacheEvict

Spring Boot automatically configures a suitable cache manager to server as a provider for the relevant cache.

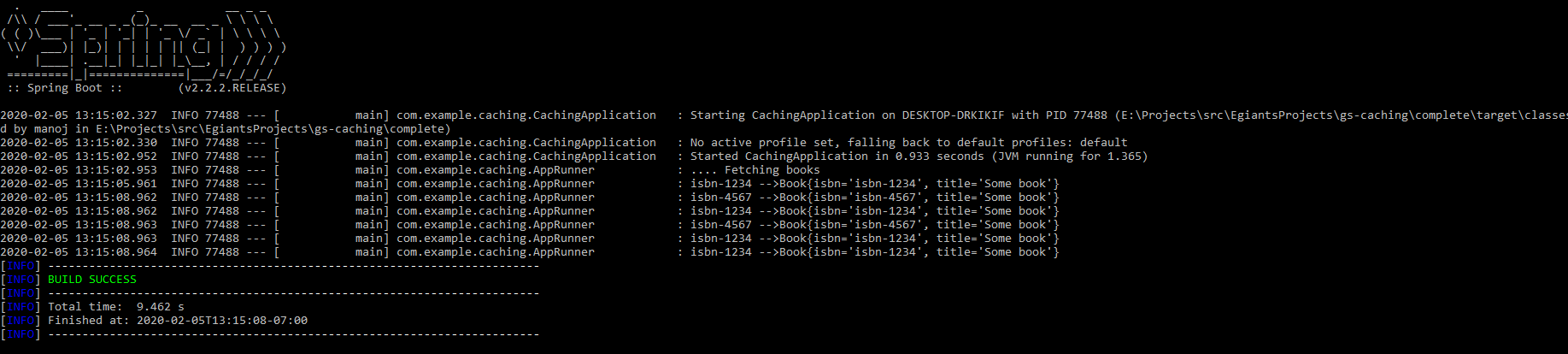
**Output:**

**Without Caching:**



Here, with out caching the application is taking 3 seconds each time to fetch each record. Because the caching is not enabled

**With Caching:**



Unlike the above example, After enabling the caching techniques, for the first time alone the application is taking 3 seconds and displaying all the records at once.

To load the first records, there is the system wait for 3 seconds that why the application took 3 seconds to display that record.

But since we enabled the cache the preprocessor already looked for @Cacheable annotation and found @Cacheable(“Books”). Then the cache Manager loads all the other records in books repository to the cache for quick access.

This may be a 3 seconds less time different for this application, but it will show a huge impact when we are loading millions of records using our application.

1. **Spring Boot Quartz scheduler**

Quartz is on of the most popular scheduling library.

There are 3 main components in Quartz

1. Job – Job is nothing but a task.

Eg: printing a statement, adding an user to database etc.

1. Trigger – this is used to trigger a task.
   1. Simple Trigger – Lifetime only once
   2. Corn trigger – I can be schedule the trigger based on time , like once in a hour like that.
2. Scheduler – to schedule a particular job within the given trigger.

In this example we are printing a statement using this quartz scheduler for every second.

We can set the frequency using the application.properties like below.

simplejob.frequency=1000

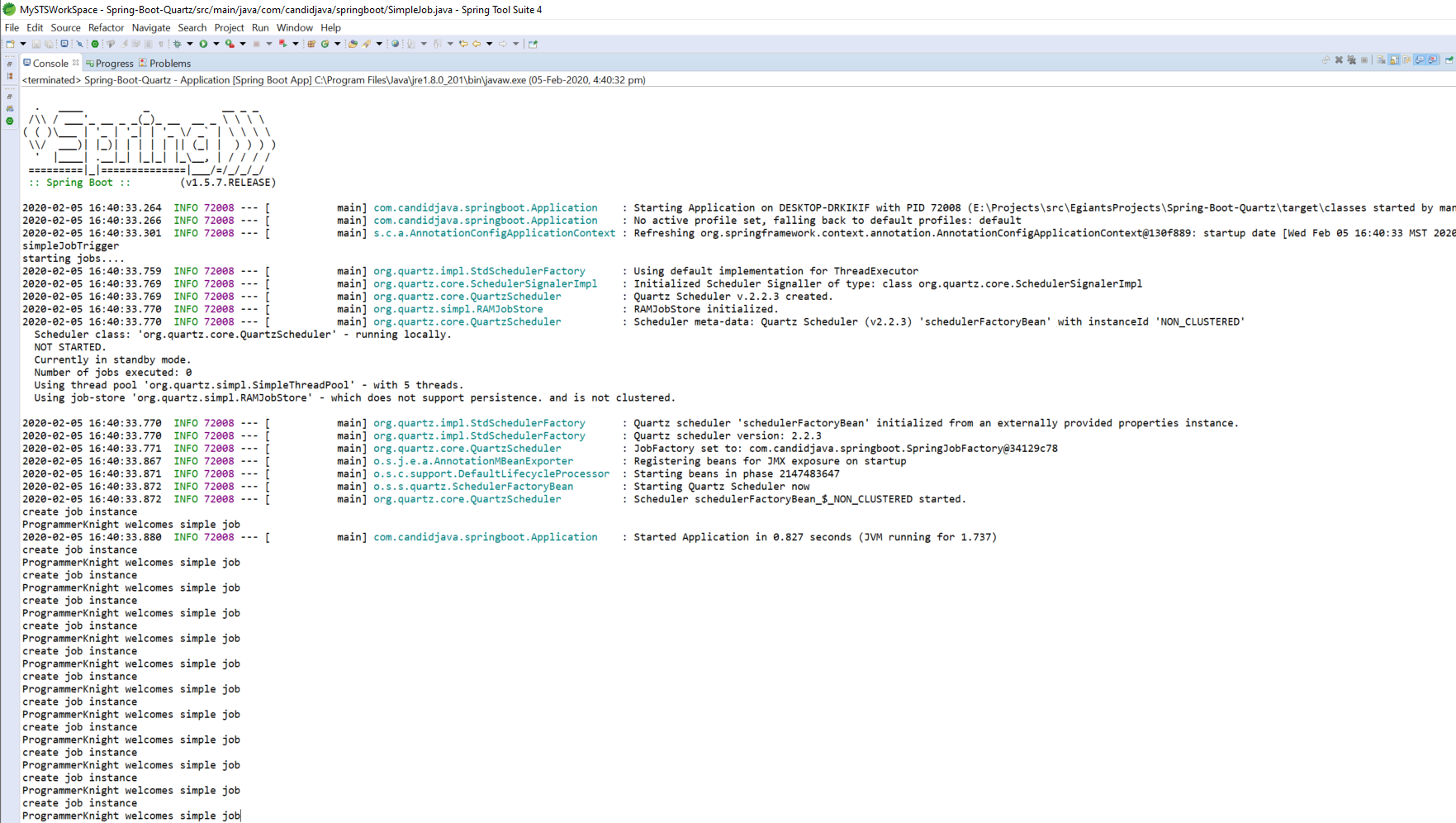
to dynamically update the value in entire code.

We can schedule the task and increase the threads to run the tasks using the quartz.properties file

org.quartz.scheduler.instanceName=springBootQuartzApp

org.quartz.scheduler.instanceId=AUTO

org.quartz.threadPool.threadCount=5



**I acknowledge that this document can be supplied to USCIS in compliance with CPT/OPT/STEM OPT audit:**

**Manoj Kumar Yekollu.**