##Lab 8 - Using Spring Cloud Bus

**Part 1 - The Broker**

1. Download Rabbit MQ from<https://www.rabbitmq.com/download.html>. Use appropriate distribution for your platform.
2. Launch Rabbit MQ and leave it running.

**Part 2 - The Server**

1. Stop ALL of the services that you may have running from previous exercises. If using an IDE you may also wish to close all of the projects that are not related to "lab-8” or "common".
2. Open lab-8-config-server. Open the POM, add another dependency for spring-cloud-starter-bus-amqp. (Notice the Spring Cloud version of Camden.SR4 - this is to avoid an issue with Camden.SR5)
3. Open application.yml. Change the spring.cloud.config.server.git.uri to your own personal git repository. If you are not sure what this is, take a look back at lab 3 where we first introduced spring cloud config.
4. Save your work and run the lab-8-config-server.

**Part 3 - The Client**

1. Open lab-8-client. Open the POM, add another dependency for spring-cloud-starter-bus-amqp. Also add a dependency for Spring Boot Actuator (org.springframework.boot / spring-boot-starter-actuator)
2. Open the LuckyWordController. Add a @ConfigurationProperties annotation using a prefix of “wordConfig”. Notice the properties / getters and setters.
3. Open bootstrap.yml. Notice the application name. What file will be needed in your Git repository to hold properties for this application?

**Part 4 - The Repository**

1. Within your Git repository, create a lucky-word-client.yml (or .properties) file. Add a level for “wordConfig:” and entries within it for “luckyWord:” and “preamble:”. Add a value for lucky word, something like “Irish” or “Serendipity” or “Rabbit’s Foot”. For preamble add a value such as “The lucky word is”. If you need a hint take a look at the “ConfigData” folder in the student files. Save your work and commit to the repository.

**Part 5 - Running**

1. Start your lab-8-client application. It should start without errors.
2. Open a browser to<http://localhost:8002/lucky-word>. You should see output such as “The lucky word is: Irish”. If you do not, review the previous steps.

**Part 6 - Configuration Changes**

1. Return to your Git repository and edit your lucky-word-client.yml file. Change the lucky word to some other values. Commit your work. Will this change be visible if you refresh<http://localhost:8002/lucky-word> right now? Why?
2. Make a POST request to<http://localhost:8001/bus/refresh>. You can do this using a “curl” command in Linux / Unix, or you can find a REST client plugin for one of your browsers that will do this (I use “Simple REST Client” on Chrome).
3. Refresh<http://localhost:8002/lucky-word>. Your changes should be visible. If so, congratulations, you have successfully used Spring Cloud Bus!

**BONUS - @RefreshScope**

1. Return to your LuckyWordController and convert to @RefreshScope. For test purposes, you can make the controller stateful by defining another String variable named fullStatement, and populate it in an init() method marked with a @PostConstruct annotation. Change the showLuckyWord() method to simply return fullStatement. Repeat the process to make a change to see if your @RefreshScope worked.