**Gemfire**

Requirement:Hands on experience on distributed cache such as Gemfire

<https://www.baeldung.com/spring-data-gemfire>

<https://grokonez.com/spring-framework/spring-data/create-springboot-gemfire-restfulapi>

<https://www.youtube.com/watch?v=miD9QwMKv48>

**Step to do**

**– Create a Spring Boot project add dependency**

<dependency>

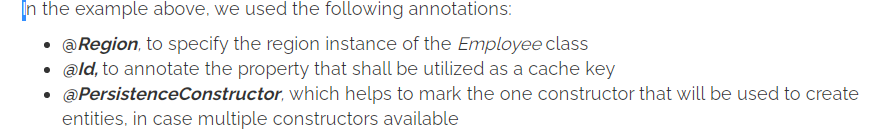
 <groupId>org.springframework.data</groupId>

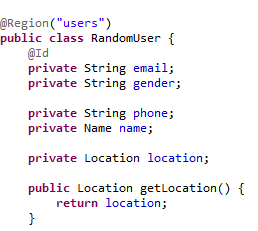
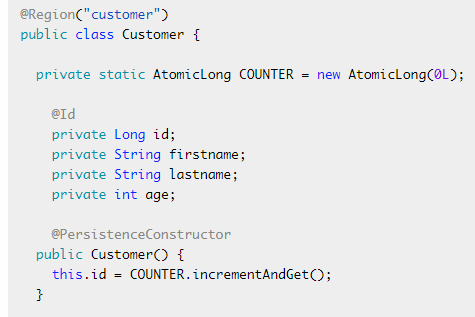
<artifactId>spring-data-gemfire</artifactId>

<version>2.3.2.RELEASE</version>

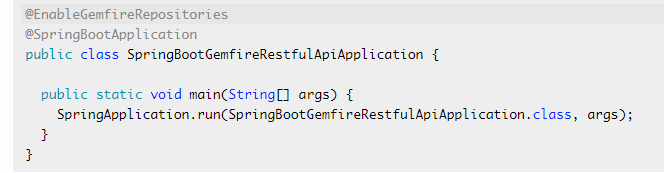
</dependency>

**– Create a Java Model class**





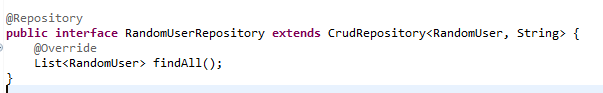
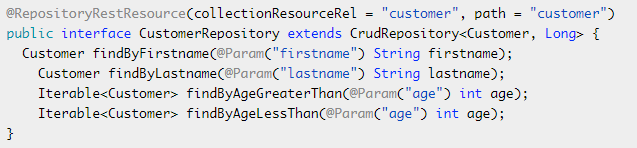
**– Config Gemfire storage**

Note: @EnableGemfireRepositories is placed in main class but We put it in config file so that code can be reuse.



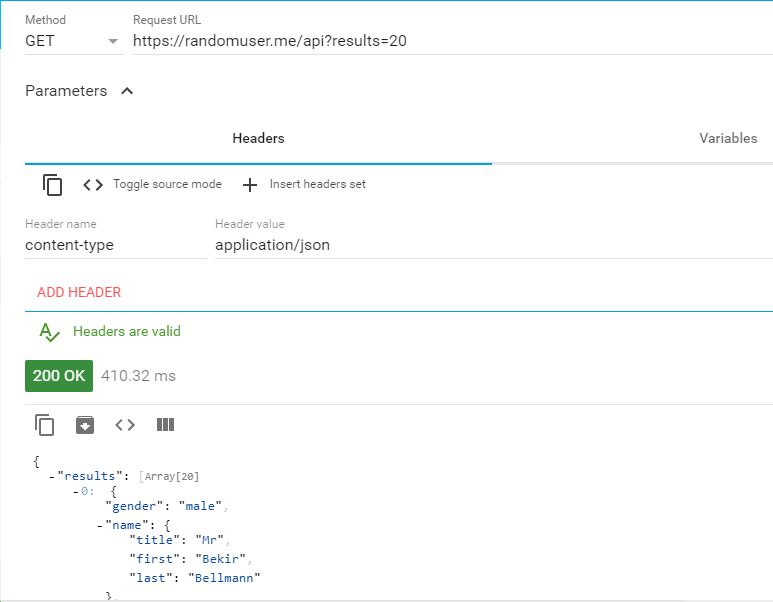
**– Create a Gemfire repository**

**@RepositoryRestResource** is optional, it is used to redefine a default path.



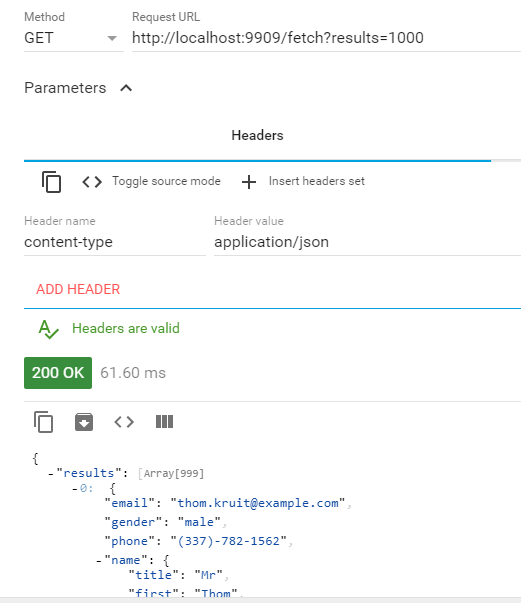
**– Run & Check result**

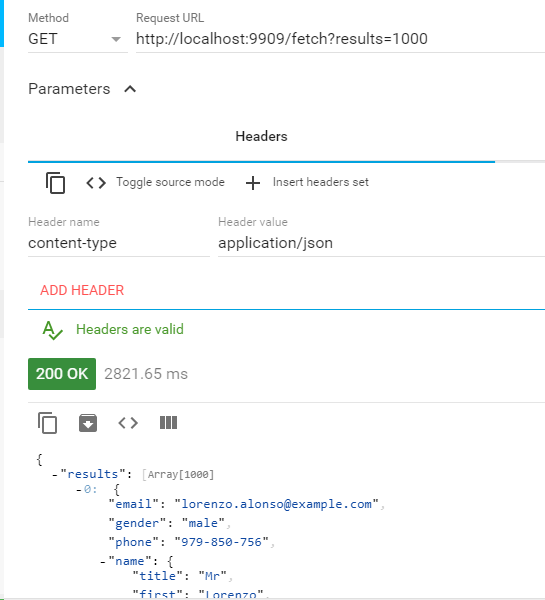
GET <https://randomuser.me/api?results=20>



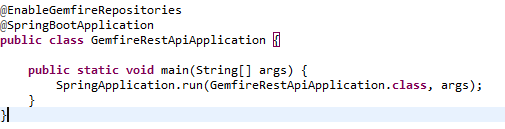
See the time before cache and after caching by gemfire

GET http://localhost:9909/fetch?results=1000

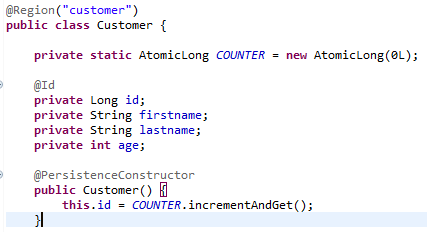


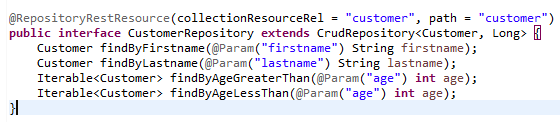


**GemfireWithRestApi**









Do these

GET <http://localhost:9090/customer>

GET http://localhost:9090/profile/customer

GET http://localhost:9090/customer/search

