Spring Cloud

* Overview
* Externalized Configuration
* Microservices
* Service Discovery
* Discovery Client
* Security
* Load Balancing
* Circuit Breaker: Resilience4j
* Zuul

Spring Cloud:

It helps in providing the tools to quickly build the distributed applications with common design patterns like:-

Service Discovery  
Discovery Client  
Circuit Breaker  
Load Balacing

Using this coordination between the distributed applications can be simplified.

With spring cloud you can simplify the development of distributed applications with simple annotations & configurations.

Spring Cloud uses the Spring Boot project to quickly develop the applications, we must use the compatible spring boot projects to work spring cloud.

Spring cloud uses many spring boot features:

* Starter Projects
* Compatible version of Spring Boot for Spring Cloud
* Creating an executable jar/war
* Property configuration
* Passing the configurations from command line at runtime
* Deploying the war on external server
* Actuators endpoints: health, metrics, beans
* MVC implementation: Service, DAO
* JPA Repository
* REST based services
* Curl commands
* Working on GIT

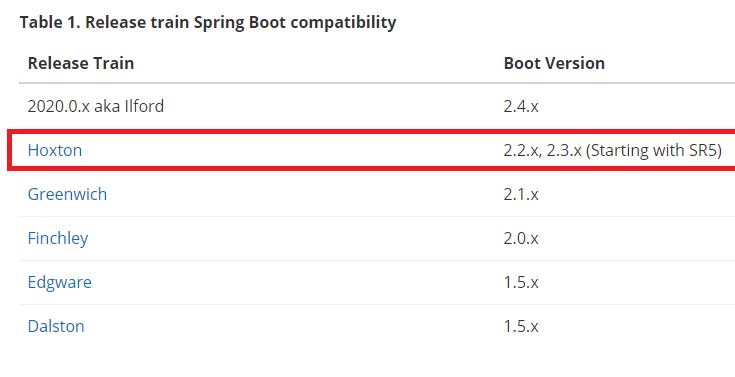
In spring cloud we are going to deal with:

* Externalized Configuration: This makes a configuration which is remotely available for multiple applications, you can apply security like Encryption by using JCE (Java Cryptographic Extension), You can also force applications to pass the credentials while accessing the remote location
* Microservices: Service Discovery, Discovery Client, Load-Balancing, we need to know some design patterns, like Development patterns, resilience pattern, routing pattern.
* Circuit Breaker: Hystrix (Deprecated), Resilience4j
* Zuul: Common door for all the incoming request to your service discovery, dynamic routing
* Secuirty

Getting Started

1. New Spring Boot project
2. Add spring cloud version to the existing project

Spring Boot compatible release train for spring cloud

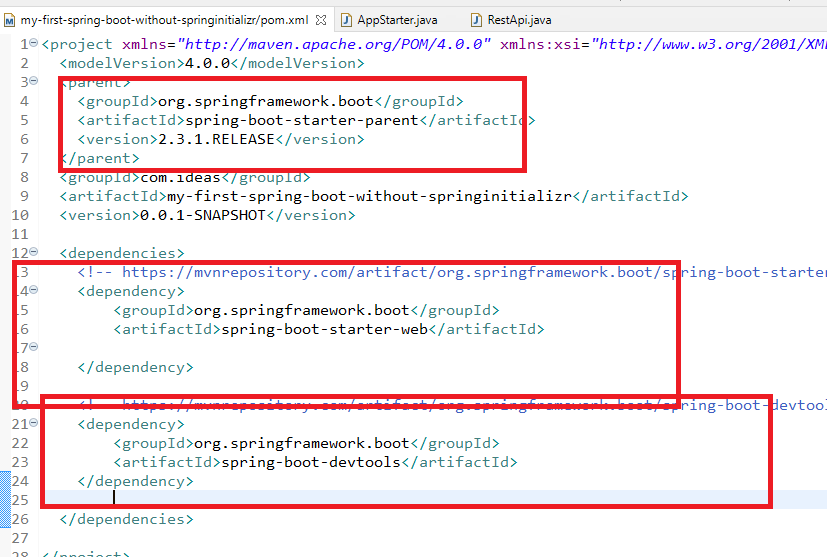


Spring Boot essentials:

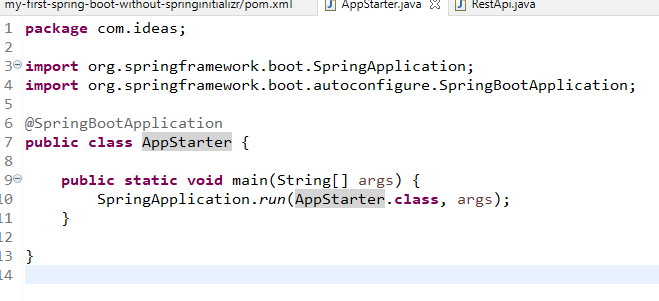
Starting from creating project to Creating rest services with JPA (in-memory database h2)

Spring Boot with Maven without spring initializr

pom.xml



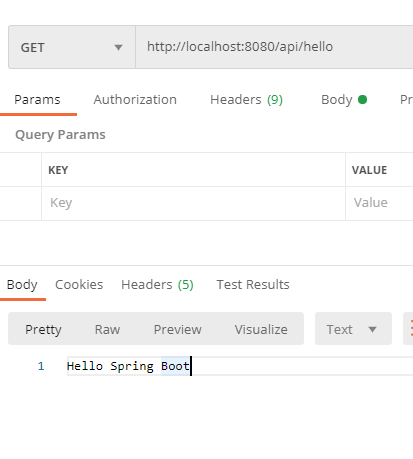
AppStarter



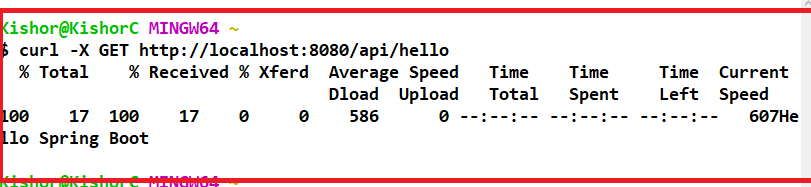
RestApi



Output:



Through cURL you can pass the GET request

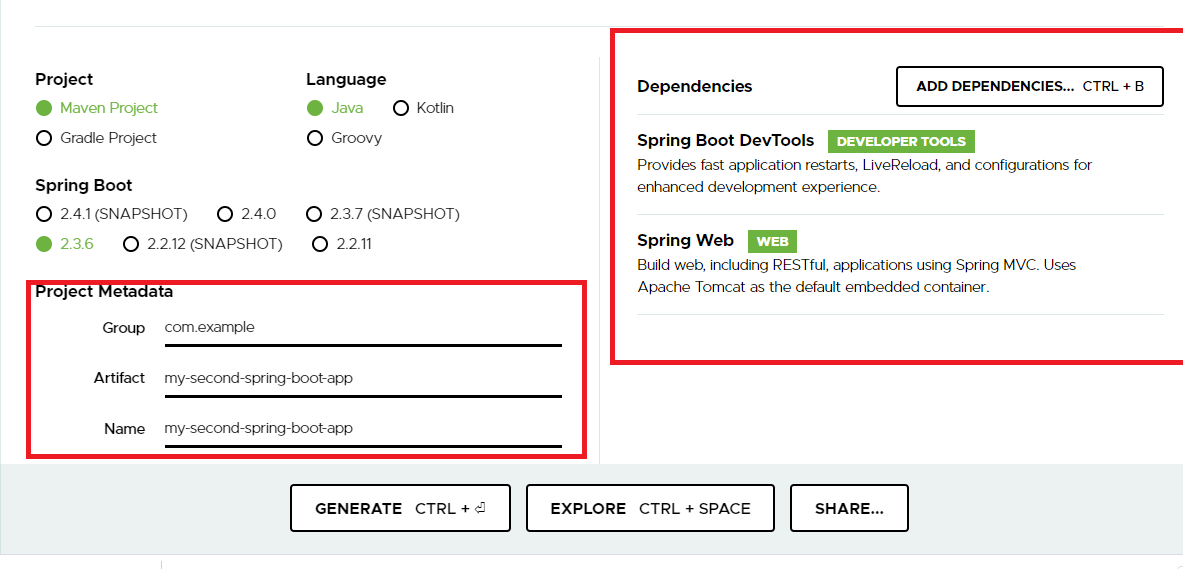


Some of the useful curl commands:

POST request with JSON data:   
curl -X POST url -H “Content-Type:application/json” -d “{...}”

PUT request:  
curl -X PUT url

Spring boot project with Spring Initializr



RestApi

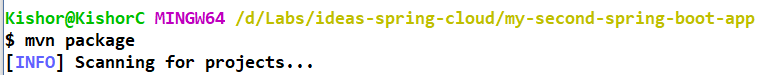


Benefits

* You will get UI based dependency configuration
* You will get pre-defined code to run/launch spring application
* You will get a plugin to create executable Jar/War
* You will get application.properties
* You will get compatible version of spring boot

How to create an executable jar file

You should use *mvn package* command



You will get a jar file inside target folder

How to execute the jar file

You should use *java -jar <<file.jar>>*

