Sprint Boot RESTful API and MongoDB

Considering some of the new points that Charlie talk me in our chats. I create a little demo with some of cool tools after quick review for the material in the Spring Boot website, using his advice and recommendations in the developer guides.

Requirements

1. JDK 1.8 SE
2. Maven 3.x
3. Spring Boot 1.2.5
4. MongoDB
5. Eclipse Mars
6. PostMan Chrome plug-in to Launch Rest

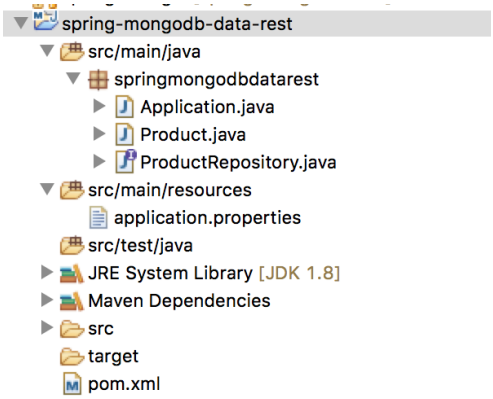
The demo is a CRUD operation across a Service RESTful using the Product concept.

1. Product creation
2. Product View
3. Product Delete

Spring Boot. Is part of the new tool of Spring framework that simplifies the development of Spring applications.

This is the structure inside Eclipse Mars

Configuration



The pom.xml contains all the dependencies. Including a Tomcat container for web service deploy

This is the starter or Boot Up for the Spring boot app and go get the dependencies and run required components

Application.java

```java
package springmongodbdatarest;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}

```

Repository

Is necessary a repository interface to make MongoDB related operations using the Product Pojo

Application.java | Product.java | ProductRepository.java

```java
package springmongodbdatarest;

import org.springframework.data.mongodb.repository.MongoRepository;

@RepositoryRestResource(collectionResourceRel = "products", path = "products")
public interface ProductRepository extends MongoRepository<Product, String> {

}

```

Product also will be considered like the model mapping into the MongoDB and Spring automatically use save() or find() actions in MongoDb.

To start in a terminal using Maven to build the app and start the app

Mvn spring-boot:run

```
SpringMongoDBDataRestExample — java -Xmx1024m -XX:MaxPermSize=512m -classpath ~/Documents/Dev/Maven/boot/plexus-classworlds-2.5.1.jar -Dclassworlds.conf=/Users/Mikauran/Documents/Dev/Maven/bin/m2.
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.2.5.RELEASE)
  Bolas
2015-11-11 21:57:30.310  INFO 6980 --- [lication.main()] springmongodbdatarest.Application        : Starting Application on Macorola.local with PID 6980 (started by Mikauran in /Users/Mikauran/Documents/Dev/Java Dev/MondoDb/Spring
ataRestExample)
2015-11-11 21:57:30.343  INFO 6980 --- [lication.main()] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@57647a2c: startup date [Wed Nov
7:30 PST 2015]; root of context hierarchy
2015-11-11 21:57:30.703  INFO 6980 --- [lication.main()] o.s.b.f.s.DefaultListableBeanFactory     : Overriding bean definition for bean 'beanNameViewResolver': replacing [Root bean: class [null]; scope=; abstract=false; lazyInit=f
towireMode=3; dependencyCheck=0; autowireCandidate=true; primary=false; factoryBeanName=org.springframework.boot.autoconfigure.web.ErrorMvcAutoConfiguration$WhitelabelErrorViewConfiguration; factoryMethodName=beanNameViewResolver;
hodName=null; destroyMethodName=(inferred); defined in class path resource [org/springframework/boot/autoconfigure/web/ErrorMvcAutoConfiguration$WhitelabelErrorViewConfiguration.class]] with [Root bean: class [null]; scope=; abstr
e; lazyInit=false; autowireMode=3; dependencyCheck=0; autowireCandidate=true; primary=false; factoryBeanName=org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfigurationAdapter; factoryMethodName=bean
Resolver; initMethodName=null; destroyMethodName=(inferred); defined in class path resource [org/springframework/boot/autoconfigure/web/WebMvcAutoConfiguration$WebMvcAutoConfigurationAdapter.class]]
2015-11-11 21:57:31.389  INFO 6980 --- [lication.main()] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat initialized with port(s): 8080 (http)
2015-11-11 21:57:31.517  INFO 6980 --- [lication.main()] o.apache.catalina.core.StandardService   : Starting service Tomcat
2015-11-11 21:57:31.518  INFO 6980 --- [lication.main()] org.apache.catalina.core.StandardEngine  : Starting Servlet Engine: Apache Tomcat/8.0.23
2015-11-11 21:57:31.575  INFO 6980 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2015-11-11 21:57:31.575  INFO 6980 --- [ost-startStop-1] o.s.web.context.ContextLoader            : Root WebApplicationContext: initialization completed in 1232 ms
2015-11-11 21:57:32.296  INFO 6980 --- [ost-startStop-1] o.s.b.c.e.ServletRegistrationBean        : Mapping servlet: 'dispatcherServlet' to [/]
2015-11-11 21:57:32.299  INFO 6980 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean  : Mapping filter: 'characterEncodingFilter' to: [/*]
2015-11-11 21:57:32.299  INFO 6980 --- [ost-startStop-1] o.s.b.c.embedded.FilterRegistrationBean  : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
2015-11-11 21:57:32.381  INFO 6980 --- [lication.main()] o.s.b.f.config.PropertiesFactoryBean     : Loading properties file from class path resource [rest-default-messages.properties]
2015-11-11 21:57:32.442  INFO 6980 --- [lication.main()] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@57647a2c: s
ate [Wed Nov 11 21:57:30 PST 2015]; root of context hierarchy
2015-11-11 21:57:32.475  INFO 6980 --- [lication.main()] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String, java.lang.Object>> org.spr
work.boot.autoconfigure.web.BasicErrorController.error(javax.servlet.http.HttpServletRequest)
2015-11-11 21:57:32.475  INFO 6980 --- [lication.main()] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autocon
eb.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest)
2015-11-11 21:57:32.490  INFO 6980 --- [lication.main()] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2015-11-11 21:57:32.490  INFO 6980 --- [lication.main()] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2015-11-11 21:57:32.511  INFO 6980 --- [lication.main()] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2015-11-11 21:57:32.554  INFO 6980 --- [lication.main()] o.s.d.r.w.RepositoryRestHandlerAdapter   : Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@57647a2c: s
```

Running and in Action

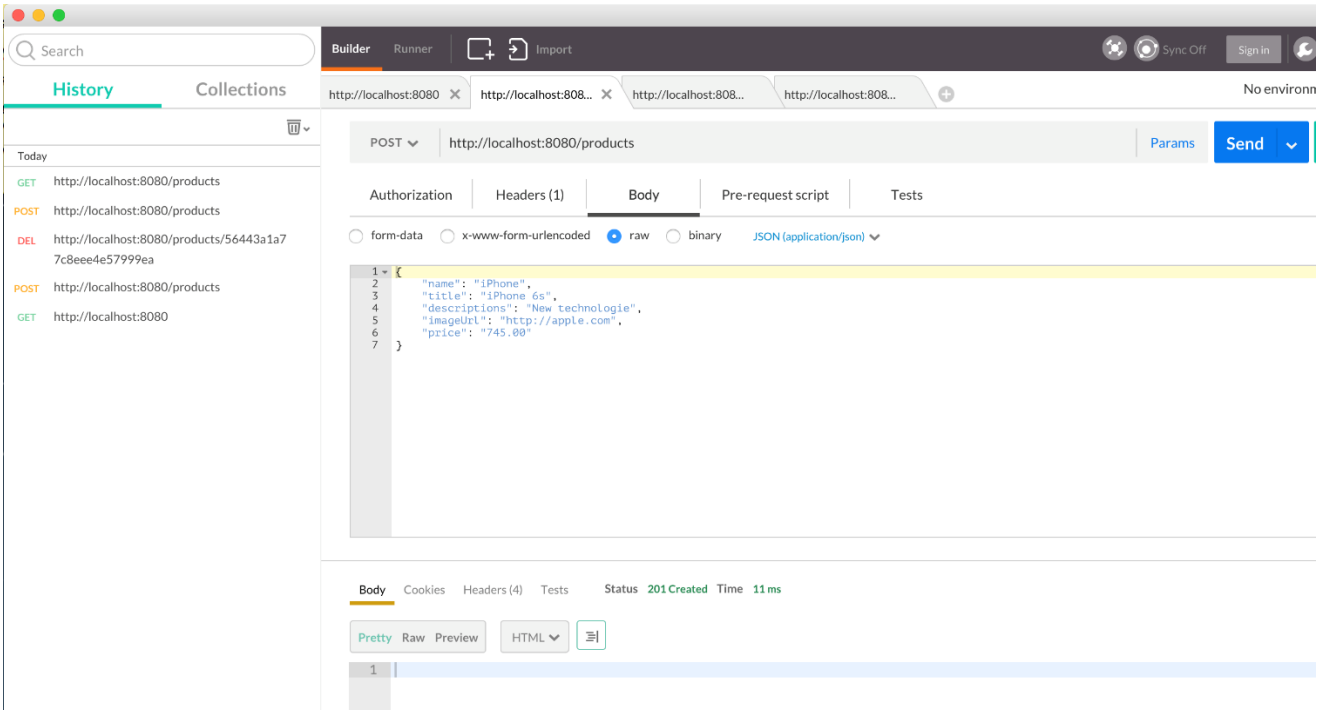
After the launching is possible invoke the RESTful service using his endpoints.

Andreas I'm using PostMan chrome plug-in to test the WebService endpoints, in the interview I comments to you that other important tool is SOAUI but this options also is cool.

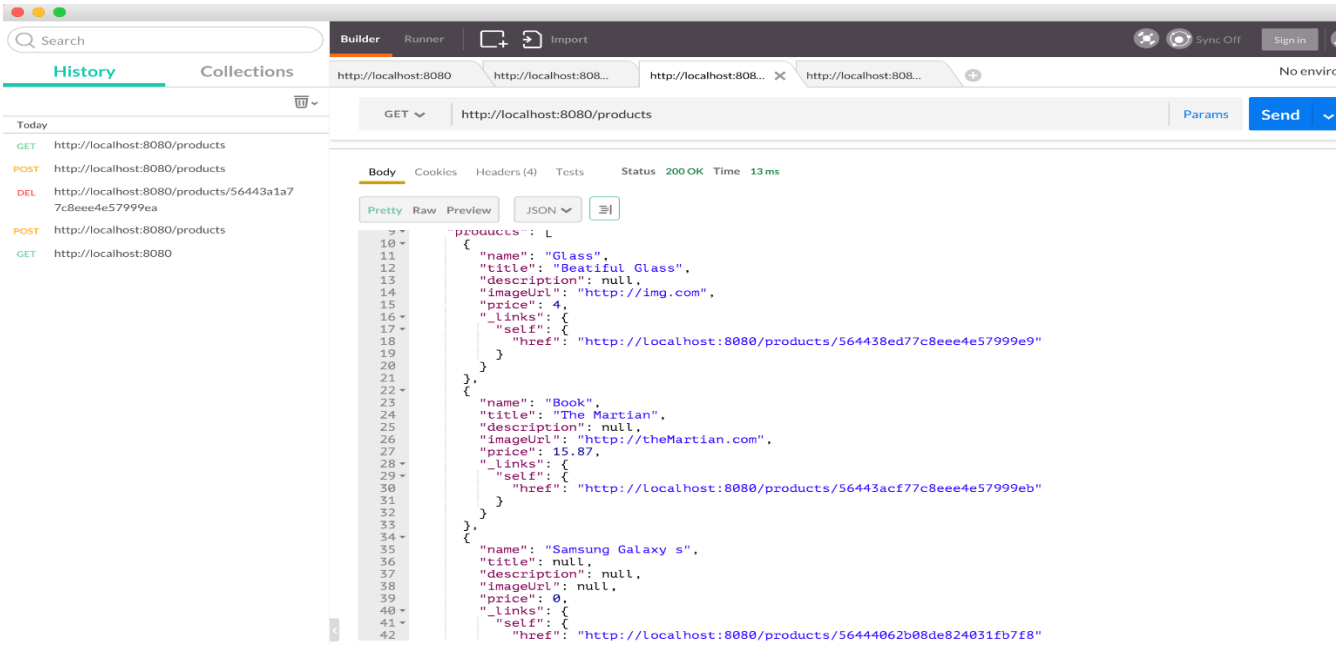
GET: View

http://localhost:8080

POST: Create



GET : View after the Product insert



DELETE: http://localhost:8080/products/56440628de824031fb7f8

Inside MongoDB

Finally for launch another test. I'm insert a Product record directly in the database

```
Mikauran — mongo — 158×24
  db.product.insert({name:"Samsung Galaxy s"},{title:"Galaxy Phone"},{descriptions:"Phone new"},{imageUrl:"http://samsung.com"},{price:"650.00"})
WriteResult({ "nInserted" : 1 })
> db.product.count()
3
> db.product.find()
{ "_id" : ObjectId("564438ed77c8eee4e57999e9"), "_class" : "springmongodbdatarest.Product", "name" : "Glass", "title" : "Beatiful Glass", "imageUrl" : "http:/
/img.com", "price" : 4 }
{ "_id" : ObjectId("56443acf77c8eee4e57999eb"), "_class" : "springmongodbdatarest.Product", "name" : "Book", "title" : "The Martian", "imageUrl" : "http://the
Martian.com", "price" : 15.87 }
{ "_id" : ObjectId("56444062b08de824031fb7f8"), "name" : "Samsung Galaxy s" }
>
```