Creation du projet depuis maven clear

1. Creation parent group pom.xml

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 <version>2.5.1</version>  
 <relativePath/>  
</parent>

<properties>  
 <java.version>1.11</java.version>  
</properties>

<dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
</dependencies>

<build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
</build>

1. Creation de la premiere classe avec la notation @SpringBootApplication (run void project spring)

import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class TodoApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(TodoApplication.class, args);  
 }  
}

* Preparation vertuel configuration par defaut
* Spring application context : container qui contient le code
* Demarrage du serveur Tomcat
* Class path scan pour voir les annotation est differencier les different service les controller les services et data access

1. Controllers avec premier fonction greeting(),greetingWithName(@PathVariable String name)

@RestController  
public class HomeController {  
  
 @RequestMapping(value = "/")  
 public String greeting(){  
 return "Hello World !!!";  
 }  
}

@GetMapping(value = "/{name}")  
public String greetingWithName(@PathVariable String name){  
 return String.*format*("welcom %s to spring boot application", name);  
  
}

1. Creation class todo et le controller pour todo

public class Todo {  
  
 private int id; …

@RestController  
@RequestMapping(value = "/api/v1/todo")  
public class TodoController {  
  
 @GetMapping(value = {"","/"})  
 public Todo listTodos(){  
 Todo test = new Todo(1,"first toto title","first todo description");  
 return test;  
 }  
}

V2 Controller Todos une liste à la place d’un seul individuel

@RestController  
@RequestMapping(value = "/api/v1/todo")  
public class TodoController {  
  
 @GetMapping(value = {"","/"})  
 public List<Todo> listTodos(){  
 return Arrays.*asList*(  
 new Todo(1,"first toto title","first todo description"),  
 new Todo(2,"second toto title","second todo description"),  
 new Todo(3,"third toto title","third todo description")  
 );  
 }

La dependence devtools pour l’automatisation du demmarage qu’on une modification ctr+f9 build project

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <version>2.5.1</version>

<scope>runtime</scope>  
</dependency>

1. Changenet au controller todo declaration liste data de type todo et le retour de data dans @GetMapping

@RestController  
@RequestMapping(value = "/api/v1/todo")  
public class TodoController {  
  
  
  
 private List<Todo> data = Arrays.*asList*(  
 new Todo(1,"first toto title","first todo description"),  
 new Todo(2,"second toto title","second todo description"),  
 new Todo(3,"third toto title","third s todo description")  
 );  
 @GetMapping(value = {"","/"})  
 public List<Todo> listTodos(){  
 return data;  
 }

Ep 2 : creation service layer et le changeent d’emplacement de données en relation avec cette couche(buseness logiq)

1. Creation classe TodoServices

@Service  
public class TodoService {  
 private List<Todo> data = new ArrayList<>(Arrays.*asList*(  
 new Todo(1,"first toto title","first todo description")…

public List<Todo> findAll() {return data;}

public Todo getById(int id) {  
 for (Todo todo : data) {  
 if (todo.getId() == id) {  
 return todo;  
 } }return null;}

public boolean save(Todo todo){  
 return data.add(todo);  
}

public void delete(int id){  
 for (Todo todo : data){  
 if(todo.getId()==id){  
 data.remove(todo);}

1. Installation database mongodb configuration fichier .yml et la <dependency>mongodb

Creation interface d’acces au données TodoRepository @Repository