Machine generated alternative text:
What is Spring Boot? 
Spring Boot makes it easy to create 
stand-alone, production-grade Spring 
based Applications that you can "just 
run". 
... create ... 
Spring based Applications ... 

Ready to deploy in production

Machine generated alternative text:
What is Spring? 
• Application framework 
• Programming and configuration model 
• Infrastructure support 

Support many db

Machine generated alternative text:
Problems with Spring 
• Huge framework 
• Multiple setup steps 
• Multiple configuration steps 
• Multiple build and deploy steps 
Can we abstract these steps? 

Machine generated alternative text:
Enter Spring Boot 
• Opinionated 
• Convention over configuration 
• Stand alone 

**NOTE :-**

**in spring boot no need to find a container just run it**

**Pom.xml :- add dependency**

**Group id :-** package name

**Artifact id** :- project name

Machine generated alternative text:
Spring • • Spring Tool Suite • /ugorWkkOthagaWDocumøntg/spnngbOOt 
n S "http:/.'m."3.Org/Z"WXMt Sch 
SRA est,'resu rces 
'RE t_bwy 
Severs 
SevVS 
— eproject 
_ javabrai springbootquickstartogrouprd» 
i fact i fact 
.ersion,ø.ø. 
Brains Course 
cpa ren 
groupld»org. . boo t Ogr•ouprd. 
«arti factId'spring-bOOt-Starter-parentUarti factIO 
cd ependenep 
cgrou 
«arti factid» 
Odependencp 
. vjavo. 
up roj ec t' 

**Our project is a child of parent project.**

**The idea is that child project will inherit features of parent project**

**Note : spring boot created a project :-** spring-boot-starter-parent and they put all default configuration in that project

===============================================================================

**Demo :-**

1. **We need to tell spring that this is a spring application( main class) will use:**

**@SpringBootApplication :** this annotation will tell to spring boot this is the starting point of application

**Note**

Need to tell springboot start the application and create servlet container and host this application in the servlet container and make it available and for this we need to call one static class

**SpringApplication.run(first arg, second arg)**

**First arg :-** is the name of class where we have main method that is annotated with @SpringBootApplication

**Second arg** :- is the argument that is passed to main method

============================================================

Every spring application has application context which run when application start

Machine generated alternative text:
• 
• 
• 
• 
Starting Spring Boot 
Sets up default configuration 
Starts Spring application context 
Performs class path scan 
Starts Tomcat server 

Default configuration :-spring-boot-starter-parent

Class path scan :-- look all the annotation , it treated them as service and controller as controller

==============================

**Query :** for a given request want to execute it on servlet controller

**How to do it ?** Will add controller

Machine generated alternative text:
Let's add a controller 
• A Java class 
• Marked with annotations 
• Has info about 
• What URL access triggers it? 
• What method to run when accessed 

Machine generated alternative text:
iO.j"obrainS.SpringboOtStarter.he110; 
org.springframenrk.web.bind.an_nOtation.ReStContrOtter•; 
PRestContr•oller 

**What is the use of @RestController?**

RestController is a Spring annotation that is used **to build REST API in a declarative way**. RestController annotation is applied to a class to mark it as a request handler, and Spring will do the building and provide the RESTful web service at runtime.

**Note:**

RestController define in spring mvc. This class has a method which can be mapped to a request

Machine generated alternative text:
The @RequestMapping maps only to the GET method by default. To map to other 
HTTP methods, you'll have to specify it in the annotation. 

When spring application start, spring will do class path scan and it will get 2 annotation and will register it

**Mapping**

1. **RestController**
2. **RequestMapping (note will get response as json automatically)**

Machine generated alternative text:
10-0". I 
. LEASE 

**Parent : told what version need to download of web**

**Note :** we have to worry of parent not individual version

Machine generated alternative text:
Embedded Tomcat Server 
• Convenience 
• Serviet container config is now 
application config 
• Standalone application 
• Useful for microservices architecture 

Machine generated alternative text:
Topics: 
GET 
GET 
POST 
PUT 
DELETE 
Course API 
/topics 
/topics/id 
/topics 
/topics/id 
/topics/id 
Gets all topics 
Gets the topic 
Create new topic 
Updates the topic 
Deletes the topic 

Machine generated alternative text:
package 
public class 
Topi C e 

Machine generated alternative text:
inport 
. util. Arrays • 
public clas 
TopicService 
return 
new "Spring *Spring Franewor•k Description-), 
"Core Description"). 
"JavaScript • 
Oescr•iptim•) 

**Note : @Service**

when application start it will scan all classes and create instance of class and register it.

It is a singleton , it will not create instance of class everytime.

Instance created by @service , it will take the instance and inject into topicService which is autowired.

Machine generated alternative text:


@PathVariable : what is expecting as id that is variable

Machine generated alternative text:
•Request"aøpingeethod-Requestgethod.posr, vade--,'topics") 
void addTopic(Aequestsody topic) 

@RequestBody :- tell spring that request payload will contain the instance of Topic ,take Topic from request and convert it into instance

Note: Arrays.asList---------is immutable

Machine generated alternative text:
lopiccontroller.java 
U lopicSen.'lce.Java 
List.class 
Profile' , produces 
U CourseAplApp.Java 
"hello 
Ion.class 
" application/ j son " ) 
public class TopicContr011er 
.5 
.6 
.8 
.99 
13 
15 
16 
@Autowired 
private TopicService topicService; 
public List' Topic) getA11Topics() 
return topicService . getA11Topics(); 
public Topic getTopic(@PathVariab1e 
String id) 
return topicService . get Topic(id); 
. POST , value:" /topics" ) 
public void addTopic(@RequestBody Topic topic) 
topicService . addTopic (topic) ; 
. PUT , value:" / topics/fidY') 
public void updateTopic(@RequestBody Topic topic 
@PathVariab1e 
String id) 
topicService. updateTopic(topic ,id); 
. DELETE , value:" / topics/fidY') 
public void deleteTopic(@PathVariab1e 
String id) 
topicService . deleÉeTopic (id) ; 

Machine generated alternative text:
9 @Service 
public class TopicSerwice 
11 
129 
13 
14 
15 
16 
179 
18 
19 
21 
22 
239 
24 
25 
26 
289 
29 
31 
32 
339 
34 
35 
36 
38 
39 
41 
42 
43 
45 
7 
new ArrayListO(Arrays .asL ist( 
List(Topic) topicSepvice 
new Topic( " spring" , " spring framework" , " spring framework description " 
new Topic( "java" 
cope Java" , " cope java descriptionl 
public List(Topic) getA11Topics 
return topicService; 
public Topic getTopic(String id) 
return topicService. stream() . filter(t-) t.getld() . equals(id)) .findFirst() .get(); 
public void add Topic (Topic topic) 
topicSepvice . add(topic) ; 
public void updateTopic(Topic topic, String id) 
for(int 
Topic t: topicSepvice . get(i); 
if (t .getld() . equals(id)) 
topicSepvice. set(i, topic); 
return ; 
public void delete Topic(String id) 
topicService. rembvelf(t t.getld() . equals(id)); 

Machine generated alternative text:
In Unit 3 
• Starting a Spring Boot App 
• Spring Initializr 
• Spring Boot CLI 
• STS IDE 
• Configuration 

Creating spring project

<https://start.spring.io/>

**Springboot common application properties.**

**Note:** it look properties file automatically

**Unit 4**

**Spring data jpa**

JPA is just **a specification that facilitates object-relational mapping to manage relational data in Java applications**. ... It provides a platform to work directly with objects instead of using SQL statements.

1. We need to save topic instance

Machine generated alternative text:
javax. 
Entity; 
javax.persistence.ld; 
Topi 
public Class 
String description) 
private String 
private String 
private String 
public cc) 
name ; 
description ; 
public id, String 
super O; 
this. id id; 
this. no—e 
this. description 
description; 
public String getld() 
return idi 
public void Setld(String id) 
this. id id; 
String 
return 
public void setNane(String 
this. næe 
public String getDescription() 
return description; 
public void setOescription(String description) 
this.descriDtion description; 

@Entity : it can convert topic instance to a row

CrudRepository : is interface which contain logic for any entity class

**First arg : entity and second is datatype of id**

Machine generated alternative text:
I _L java 
package springbootstarter.topic; 
String* 
public i nterface Topi 
ry extends CrudRepository«Topic, 

**updateTopic** :- No need a separate method

Machine generated alternative text:
daw 
to. javabroins.springbootstorter.topic; 
* import java.utiI.ArrayList;D 
class TopicSerWice 
"u r ed 
private TopicRepOSitory topicRepository; 
public List«TopiC» getAlITopies() 
topics ArrayListO(); 
top i cReposi findAl I C) 
. forEoch(topiCS: : add); 
return topics; 
public Topic getTopiCCStr•ing id) 
return topicReøository.findOne(id); 
public void addTopic(Topic topic) 
top i cReposi save(topi c) ; 
public updateTOpiC(String id, 
topicReposi save(topi e) ; 
public void deleteTopic(String id) 
top d); 
topic) 

Machine generated alternative text:
You don't have to implement the method! Just declare the method with the 
findByProperty name format, and Spring Data JPA will implement the method for 

Machine generated alternative text:
Explore r 
iO.iavab'Sns 
i o spring 
java 
Topic 
TopicControlIe"av-a 
TopicSeMce java 
System Library 
Maven 
Serves 
• CourseHOPOSitOry.java 
package 
•S java.utiI.LiSt;a 
public interface CourseRepository extends CrudRepositorycCourse, 
topicld) ; 
publ c SO 
java brains. io 
12 
Make sure you always write the method name in camel-case! 
8:32 / 14:55 
cc 

Note: we are looking for topicId which is a string that’s why will use findByTopicId

1----

Machine generated alternative text:
@RestContr011er 
public class 
@Autowired 
private CourseService courseSepvice; 
@RequestMapping( " / topics/ -fid h/ courses " ) 
public List(Course) 
String id) 
return courseSerwice . getA11Cburse9(id) ; 
@RequestMapping( " / topi cs/ ftopi / courses/ -f i co " ) 
public Optional(Course) getTopic(@PathVariab1e 
String id) 
return courseService . getCourse(id) ; 
. POST , value:" / topics/ftopicldh/coupses " ) 
public void addTopic(@RequestBody 
String topicld) 
@pathVariab1e 
Course course 
course . setTopic(new Topic(topicld 
courseSepvice . addCourse ( course 
. PUT , value: " / topics/ ftopicldh/coupses,/-tico " ) 
String id) 
@pathVariab1e 
String topicld, @PathVariab1e 
" / topics/ ftopicldh/ courses/ -f i co " ) 
public void updateTopic(@RequestBody 
Course course 
course. setTopic(new Topic(topicld 
courseSepvice . updateCourse (course 
value: 
. DELETE 
public void deleteTopic(@PathVariab1e 
String id) 
courseSepvice . deleteCourse(id) ; 

2-----

Machine generated alternative text:
@Service 
public class CourseSerwice 
VS, @Autowired 
CourseRespository courseRepository; 
public List(Course) getA11Courses(String topicld) 
new ArrayListO(); 
List(Course) courses 
courseRepository. findByTopbcId (topicld) . forEach ( courses : : add) ; 
return courses; 
public Optional(Course) getCourse(String id) 
return courseRepository. findById(id) ; 
public void addCourse(Course course) 
courseRepository. save ( course 
public void updateCourse(Course course) 
courseRepository. save ( course 
public void deleteCourse(String id) 
courseRepository. deleteById(id) ; 

3----

Machine generated alternative text:
public interface CourseRespository extends CrudRepository(Course, String) 
String topicld) ; 
public List(Course) 
findByTopicId 

4-------

Machine generated alternative text:
@Entity 
public class Course 
private String id; 
private String name; 
private String description; 
String description, 
String topicld) 
@ManyT00ne 
private Topic topic; 
public Course 
public Course(String id, 
super(); 
this . id 
id; 
this . name 
name ; 
String name, 
this . description 
description ; 
new Topic(topicld, 
this . topic 
public String getld() 
return id; 
public void setld(String id) 
this . id 
id; 
public String getName 
return name; 
public void setName(String name) 
this . name 
name ; 
public String getDescription 
return description; 
public void setDescription(String description) 
this . description 
description ; 
public Topic getTopic() 
return topic; 