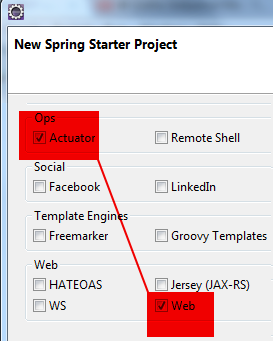
**Build a Hello World REST service in less than 6 minutes**   
<https://www.youtube.com/watch?v=47xNBNd-LLI>  
**https://www.youtube.com/watch?v=47xNBNd-LLI  
  
SHORT HAND CONVENTIONS:**  
PE == Project Explorer.  
RC == Right Click  
[ Button ] <--name of button you must click.  
Main Dialog Name: Dialog Step Name  
Right Click -> Option -> Option  
Some / Path / probably / in / Project Explorer  
  
Details: I think I will use “DemoDan” as the project name.  
  
@0:43: New --> Spring Starter Project  
@0:46: New Spring Starter Project:  
OPTIONS:  
Style:   
[X]Web (Embedded web container, which is tomcat)  
[X]Actuator (Actuator Libraries. Gives self-health metrics)  
  
Note: In updated IDE, Web+Actuator is on NEXT page.Here in photo:  
  
  
@1:34: NEXT  
Site Info:Base Url: <http://start.spring.io/starter.zip>  
Full Url: Stuff.......  
Working Sets, NOT CHECKED.  
@1:50: FINISH  
@1:56: PE/Root RC-->Run As --> 3: Java Application.  
@2:05: DIALOG: Select Java Application  
@2:09: Selected “Application – demo  
@2:10: [ OK ]  
@2:17: Whitelabel Error Page  
@2:27: Demo Application.java   
 Created for you to help you test out the process.  
@2:23: PE / src / demo (package) RC--> New --> Class  
@2:36: Dialog: Java Class  
Source folder: demo/src/main/java  
Package: demo  
Name: Greeting  
Stub methods to create: [x] Inherited abstract methods.  
@2:41[FINISH]  
@2:41-2:57: Maven Dependencies +ETC have been set up for you!  
@3:00: Inside: Greeting.java  
@3:09: @Controller annotation lets us know this is going to be a springMVC controller.  
@3:35: Return “Hello”  
@3:42: Tell it what URL to react to with:  
Annotation: @Requestmapping(“/greeting”);  
@4:06-@4:12: “The very last thing to do, is add a response body”.  
@4:16-4:18: “And that will make sure the correct content type is there”  
Annotation Referring to: @ResponseBody  
@4:22: “Great! Now we are all set!”

|  |
| --- |
| package demo; import org.springframework.stereotype.Controller; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.ResponseBody;  @Controller public class Greeting{  @RequestMapping(“/greeting”)  public @ResponseBody String greeting(){  return “Hello From Greeting.java!”;  } } |

@4:23-4:26: Hit save and re-run the service by clicking this.

|  |
| --- |
| NOTE: I am getting the “Select Java Application” thingy when I try to run now. Accordint to SpringDeveloper, this should not be happening, but we just need to select whatever class should be our MAIN class. SOLUTION AT: @2:09:  As for this solution, I am confused how the “package” is used to define the application. How does it know where the main entry point is? |
|  |

|  |
| --- |
| BUG:  Exception in thread "main" java.lang.IllegalStateException: Tomcat connector in failed state  <http://stackoverflow.com/questions/23432651/terminating-mvn-spring-bootrun-doesnt-stop-tomcat>  Basically, you need to stop your process before you run it again. |

@4:29: WhiteLabel Error Page.  
@4:32-4:34:localhost:8080/greeting  
  
**Extra cool stuff for free:**  
@4:50: localhost:8080/env <--for environment information.  
@4:59: localhost:8080/health  
@5:05: localhost:8080/beans/  
@5:11: localhost:8080/info  
@5:16: localhost:8080/metrics  
@5:22: localhost:8080/trace  
@5:28: localhost:8080/dump  
@5:37: localhost:8080/shutdown  
  
@6:08: No bloat! KTHNXBYE!

A commenter says this is NOT a rest service, maybe look into this:  
https://spring.io/guides/gs/rest-service/