<http://www.springboottutorial.com/>

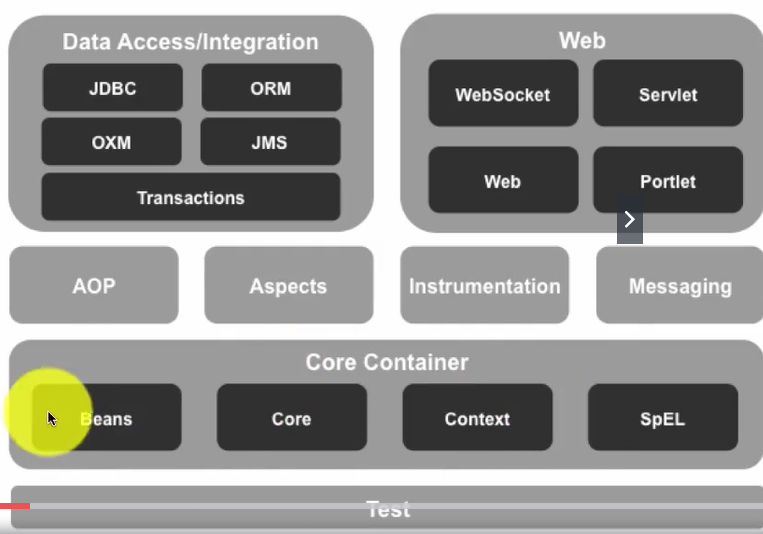
<https://github.com/in28minutes/jpa-with-hibernate>

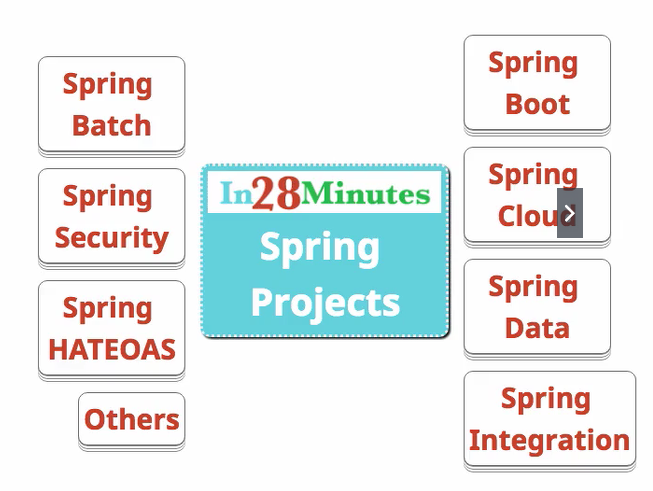
<https://github.com/in28minutes/jpa-with-hibernate#installing-tools>

<https://start.spring.io/> Spring Initializer - Create Spring projects very quickly

* Make sure to select Web to also include spring starter web in your application

Continue with <https://discovery.udemy.com/course/hibernate-jpa-tutorial-for-beginners-in-100-steps/learn/lecture/7907700#overview>





Spring Boot – Popular for developing microservices

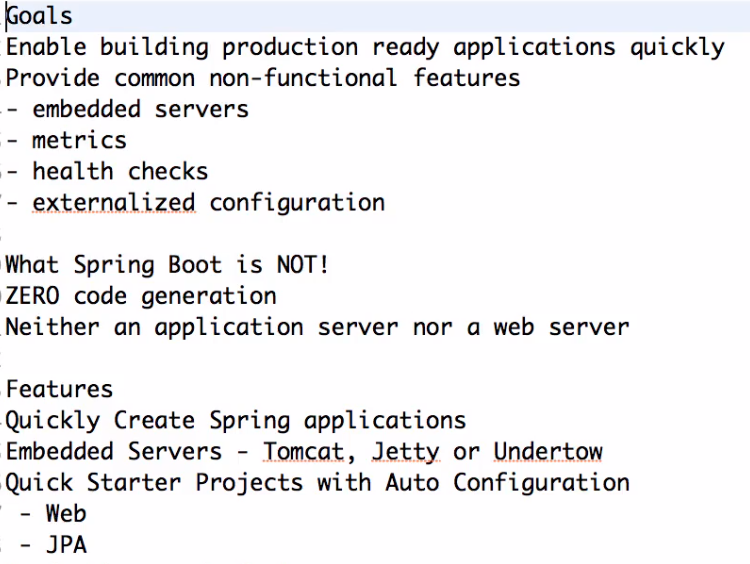
# Section 2: Introduction to Spring Boot in 10 Steps

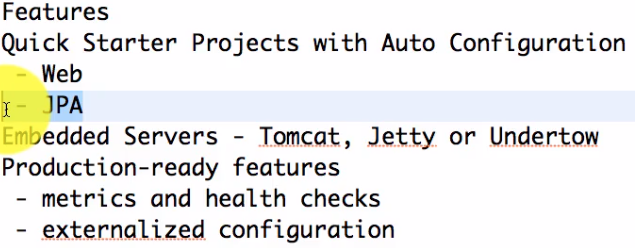
## 8. Introduction to Spring Boot in 10 Steps

* Github for this section: <https://github.com/in28minutes/jpa-with-hibernate/tree/master/00-framework-tool-introductions/02.Spring-Boot-Introduction-In-10-Steps>

## 9. Step 1: Introduction to Spring Boot – Goals and Important Features

* SpringBoot is used to create microservices in Java





SpringBoot Actutor – to monitor your application

Externalized Configuration

## 10. Step 2: Developing Spring Applications before Spring Boot

## 11. Step 3: Using Spring Initializr to create a Spring Boot Application

Go to “start.spring.io” in a browser

## 12. Fastest Approach to Solve All Your Exception

<https://github.com/in28minutes/in28minutes-initiatives/blob/master/The-in28Minutes-TroubleshootingGuide-And-FAQ/quick-start.md>

## 13 Step 4: Creating a simple REST Controller

## 14 Step 5: What is Spring Boot Auto Configuration

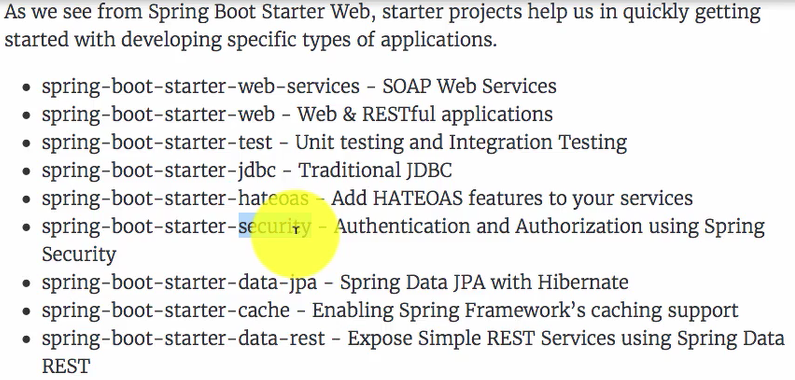
<http://www.springboottutorial.com/>

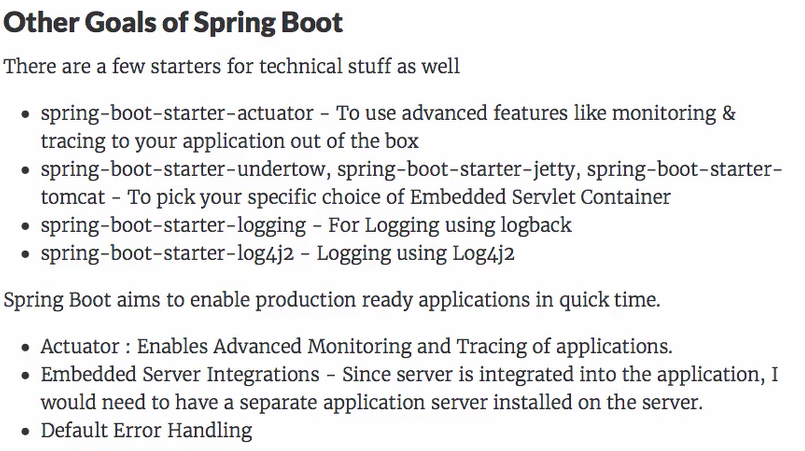
Autoconfiguration is fundamental to SpringBoot, and why it’s so easy to use.

To enable automatic debugging:

* Open application.properties
* Add the following line: logging.level.org.springframework = DEBUG

## 17 Step 8: Overview of different Spring Boot Starter Projects





<https://docs.spring.io/spring-boot/docs/2.1.7.RELEASE/reference/html/index.html>

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Google Spring Boot Reference Guide

## 18 Step 9: Spring Boot Actuator

To launch actuator

Access actuator: <http://localhost:8080/actuator>

Access HAL browser: <http://localhost:8080>

## 19 Step 10: Spring Boot Developer Tools

To automatically load changes you make to your code, while your application is running.

# Section 3: Journey from Spring JDBC to JPA

## 20. Introduction to this section

Repository for this section: <https://github.com/in28minutes/jpa-with-hibernate/tree/master/01-from-jdbc-to-jpa>

## 22. Launching H2 console

* Download and install H2 from <http://www.h2database.com/html/download.html>
* Windows key, H2 Console,
* Make sure to connect to localhost:8082/h2-console

## 23. Creating an H2 table

* In Eclipse, under resources, right click and type “SQL file”
* Create a file called data.sql. Every time you start your spring application, the H2 database will be initialized using the SQL statements in this file.
* Type the following SQL statement in data.sql

**create** **table** person

(

id **integer** **not** **null**,

name **varchar**(255) **not** **null**,

location **varchar**(255),

birth\_date **timestamp**,

**primary** **key**(id)

);

# Repositories

Main Repo for the course: <https://github.com/in28minutes/jpa-with-hibernate/>

Troubleshooting Guide: <https://github.com/in28minutes/in28minutes-initiatives/tree/master/The-in28Minutes-TroubleshootingGuide-And-FAQ>

Section 19: <https://github.com/in28minutes/jpa-with-hibernate/tree/master/00-framework-tool-introductions/01.Spring-Introduction-In-10-Steps>

Section 2: <https://github.com/in28minutes/jpa-with-hibernate/tree/master/00-framework-tool-introductions/02.Spring-Boot-Introduction-In-10-Steps>

Section 3: <https://github.com/in28minutes/jpa-with-hibernate/tree/master/01-from-jdbc-to-jpa>

# Next Udemy Course

<https://discovery.udemy.com/agile-strategy-execution/>

# Course comment

This is a note, more than a question. I created a fresh SpringBoot project using SpringInitializer for Section 3 of the course. When I attempted to run the project, I got an error. Here's portions of the stacktrace:

org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in class path resource [org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaConfiguration.class]: Invocation of init method failed; nested exception is java.lang.NoClassDefFoundError: net/bytebuddy/NamingStrategy$SuffixingRandom$BaseNameResolver

at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1796) ~[spring-beans-5.2.2.RELEASE.jar:5.2.2.RELEASE]

...

Caused by: java.lang.NoClassDefFoundError: net/bytebuddy/NamingStrategy$SuffixingRandom$BaseNameResolver

...

Caused by: java.lang.ClassNotFoundException: net.bytebuddy.NamingStrategy$SuffixingRandom$BaseNameResolver

at java.net.URLClassLoader.findClass(Unknown Source) ~[na:1.8.0\_212]

...

So in order to resolve this, I had to add the following maven dependency:

<dependency>

<groupId>net.bytebuddy</groupId>

<artifactId>byte-buddy-dep</artifactId>

<version>1.10.6</version>

</dependency>

SpringBoot now starts up, however, I am unable to access http://localhost:8080/h2-console

# Section 4 Introduction to Junit in 5 Steps

https://github.com/in28minutes/jpa-with-hibernate/tree/master/00-framework-tool-introductions/03.JUnit-Introduction-In-5-Steps

# Section 5 – Step 5 JPQL – Basics

Select c From Course c

Test