Readme.pdf

NOTE
All instructions and files are relative to package build and must be at the minimum placed with these constraints.
If you wish to dissect Lead first try reading this document from the bottom up!
(You will definitely grasp spring better from ground up this way)

This is meant to compact a few hours of Spring into hopefully 30 minutes of brief exploration but this step should indicate the very nature of Spring thus far which are it's frameworks that can either work for you or cause interference.
To run Lead Service read application.properties and configure Datasource.

Important folders to pay attention to:
src/main/java
src/main/resources

root

Defined structure of your service:
Lead.java src/main/java
LeadController.java src/main/java
Springboot Application 'Runner':
Application.java src/main/java
Repository Structure for database:
LeadRep.java src/main/java
Application Configuration:
application.properties src/main/resources

Spring Project Configuration:
pom.xml root

----------FOLLOW----------

NOTE
What you need to know:
Spring.io hosts a range of up-to-date libraries and frameworks and the lot.
For springboot, we go to their main initializer that creates a package for us/
To understand what each framework does Spring has in depth documentation at your disposal.

**You do not need to review each library to understand its uses once you understand the underlying usage

https://start.spring.io/

Visit ^^ and select to generate a Maven project with Java on 2.0.3 Spring

So far this allows us to generate a generic Maven Java project with Spring with which we choose our Dependencies and package structure.

NOTE**

The dependencies I used are given below:

spring-boot-starter-data-jpa

spring-boot-starter-web

spring-boot-starter-test

spring-boot-devtools

spring-jdbc

Each have their various uses, however special frameworks such as h2 and jdbc can interfere with each other if not careful.
This will not cover the variety of libraries to import though if you use the ones above you can infer a basic starting recommended if you begin building from scratch.

Look over to Dependencies and Select "Required" dependencies.

Choose your group Package name as well as your Artifact name.

Generate project.

IDE:

If you have Eclipse or IntelliJ (or others) build your maven project and let it configure all its dependencies (give it a few seconds)

NOTE** IntelliJ is compact futuristic and very friendly for coding in Java however I use Eclipse.

If you have problems with IntelliJ check your versions of spring, intelliJ, Java and see if there are discrepancies there.

Run on eclipse and building a maven project

Run to test your packaging and

At this point you have now successfully started your server at your localhost:8080 address

Spring packaging makes it very easy to ignore many details

----------Less Reading More Running ----------

What you should be able to do at this point:

View and understand each of the recourses provided in the Lead example.

NOTE** In each file the structure is very self explanatory and need not any explanation.

However although the general structure (Controller, Bean, Rep, Application) is very defined the rest of (layout of Beans, Controller properties) can be drastically revolutionized depending on creativity.

Lead is only one generic way of understanding your web-service.

Final Note:

Application, Repository, !!Database!!

Both the Application and your Repository have been basically handed to you and will follow your project properties as is.

Your Database can be in MySQL which is used in Lead, or follow others as long as your configuration and application.properties is good.

This does not fully define how the database is integrated with Lead however if you View:

Application.Properties

Datasource.*

You should get a clue

You should be able to understand from the properties file how your project is configured. As in Lead, the database information as well as the Config is rather simple.

IMPORTANT**

Application.Properties can be very in depth but do not overlap structures that interfere with each other.