RESTFul Web Services using SpringBoot

1. REpresentational State Transfer (REST)
2. WebService (XML/JSON)
3. SpringBoot is Opinionated framework
4. Convention over Configuration approach
5. Maven/Graddle (Build/Dependency Management Tool)
6. To create Spring Boot Application use Start.spring.io (Spring Initializr) or use STS

server.port=8082

spring.datasource.driver-class-name= com.mysql.jdbc.Driver

spring.datasource.url = jdbc:mysql://localhost:3306/sakila?useSSL=false

spring.datasource.username = root

spring.datasource.password = P@ssw0rd

# ===============================

# = DATA SOURCE

# ===============================

# Set here configurations for the database connection

# Connection url for the database "sakila"

# Keep the connection alive if idle for a long time (needed in production)

spring.datasource.testWhileIdle = true

spring.datasource.validationQuery = SELECT 1

# ===============================

# = JPA / HIBERNATE

# ===============================

# Use spring.jpa.properties.\* for Hibernate native properties (the prefix is

# stripped before adding them to the entity manager).

# Show or not log for each sql query

spring.jpa.show-sql = true

# Hibernate ddl auto (create, create-drop, update): with "update" the database

# schema will be automatically updated accordingly to java entities found in

# the project

spring.jpa.hibernate.ddl-auto = update

# Naming strategy

spring.jpa.hibernate.naming-strategy = org.hibernate.cfg.ImprovedNamingStrategy

# Allows Hibernate to generate SQL optimized for a particular DBMS

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect