Spring REST Hands-on Exercises - Answers

**Hands-on 1: Create a Spring Web Project using Maven**

Steps:  
1. Use https://start.spring.io to create project:  
 - Group: com.cognizant  
 - Artifact: spring-learn  
 - Dependencies: Spring Boot DevTools, Spring Web  
2. Main class example:  
 @SpringBootApplication  
 public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 System.out.println("Application started...");  
 }  
 }

**Hands-on 2: Load SimpleDateFormat Bean from XML**

XML (date-format.xml):

<bean id="dateFormat" class="java.text.SimpleDateFormat">  
 <constructor-arg value="dd/MM/yyyy" />  
</bean>  
  
  
Java:  
ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
Date date = format.parse("31/12/2018");  
System.out.println(date);

**Hands-on 3: Incorporate Logging**

application.properties:  
logging.level.org.springframework=info  
logging.level.com.cognizant.springlearn=debug  
logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%thread|%5p|%logger{25}|%M|%m%n  
Java:  
private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
LOGGER.info("START");  
// ...  
LOGGER.debug(date.toString());  
LOGGER.info("END");

**Hands-on 4: Load Country from XML**

country.xml:  
<bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
</bean>  
  
Country.java:  
public class Country {  
 private String code, name;  
 private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);  
 public Country() { LOGGER.debug("Inside Country Constructor."); }  
 public void setCode(String code) { LOGGER.debug("Setting Code"); this.code = code; }  
 public void setName(String name) { LOGGER.debug("Setting Name"); this.name = name; }  
 public String getCode() { LOGGER.debug("Getting Code"); return code; }  
 public String getName() { LOGGER.debug("Getting Name"); return name; }  
 public String toString() { return "Country [code=" + code + ", name=" + name + "]"; }  
}  
Java:  
ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
Country country = context.getBean("country", Country.class);  
LOGGER.debug("Country: {}", country.toString());

**Hands-on 5: Singleton vs Prototype Scope**

Singleton:  
Country country1 = context.getBean("country", Country.class);  
Country country2 = context.getBean("country", Country.class);  
// Constructor logs only once  
Prototype scope:  
<bean id="country" class="com.cognizant.springlearn.Country" scope="prototype">  
Constructor logs twice.

**Hands-on 6: Load List of Countries**

country.xml:  
<bean id="in" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
</bean>  
<!-- Similar beans for US, DE, JP -->  
  
<bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="de"/>  
 <ref bean="jp"/>  
 </list>  
 </constructor-arg>  
</bean>  
  
Java:  
List<Country> countries = context.getBean("countryList", List.class);  
for (Country c : countries) {  
 LOGGER.debug(c.toString());  
}