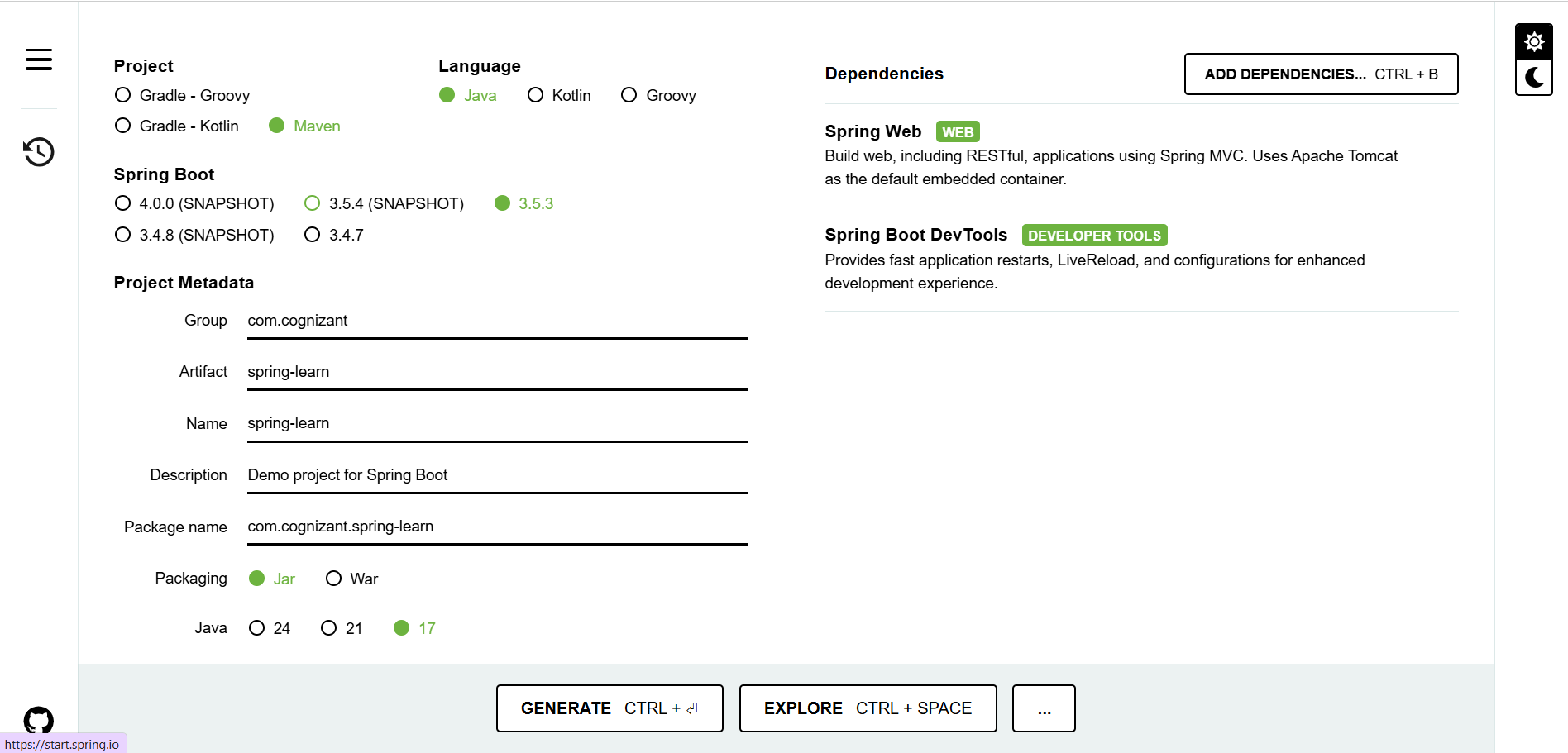
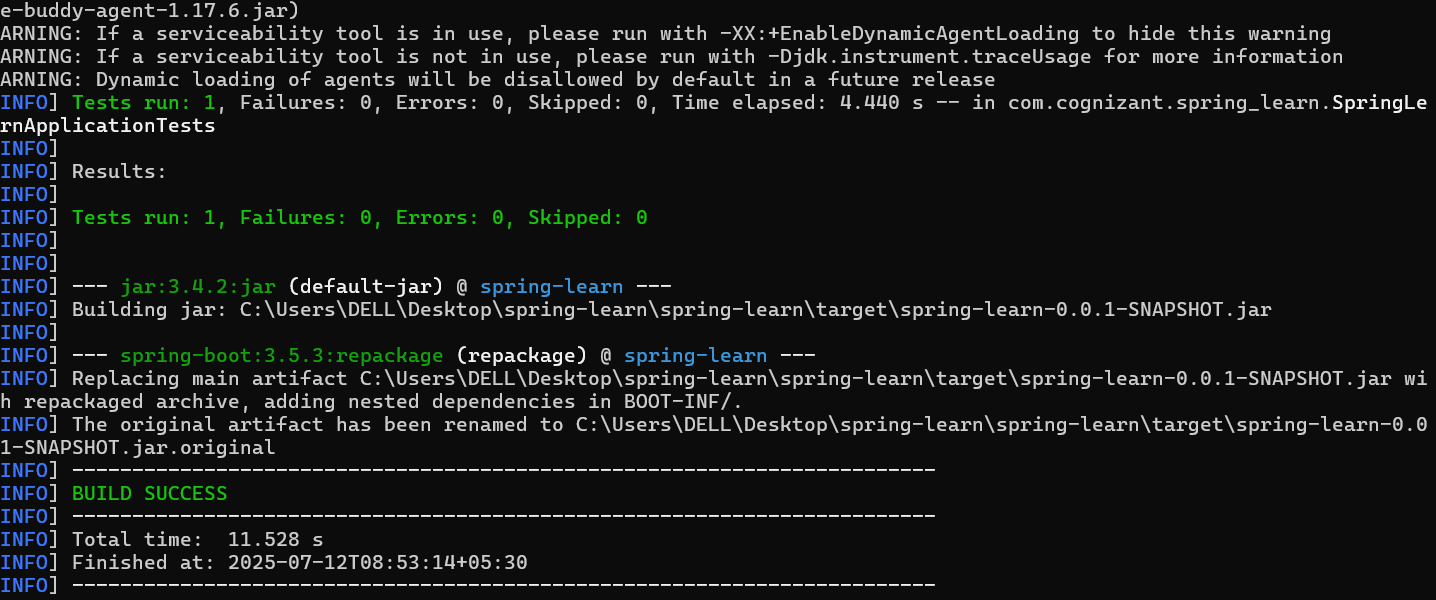
**1-SPRING-REST-HANDSON**

**Hands-on 1: Create Spring Boot Project**

**Step 1: Spring Initializr Configuration**

**STEP 2-DOWNLOAD AND EXTRACT ZIP  
STEP 3- Build with Maven**

**Step 4: Explore SpringLearnApplication.java**

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

public static void main(String[] args) {

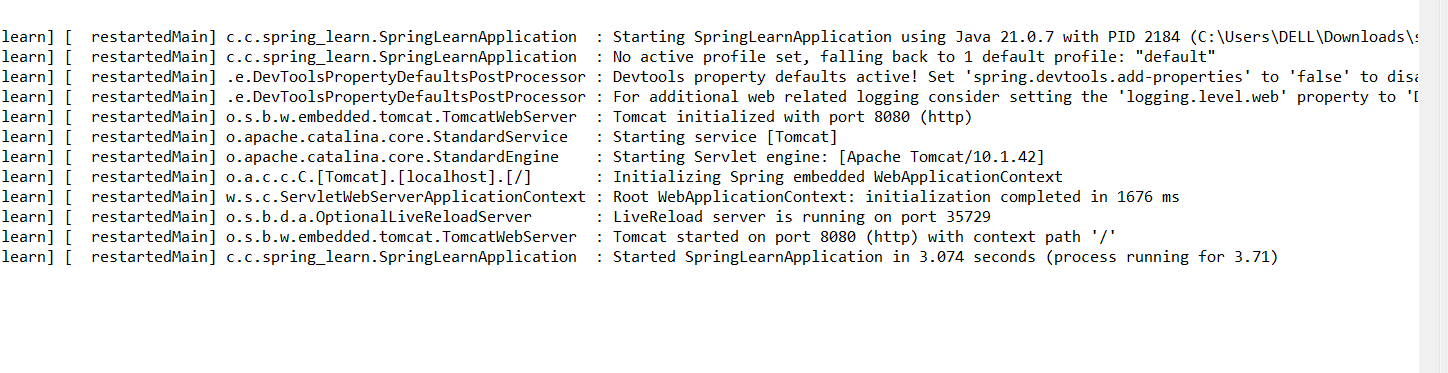
LOGGER.info("Application Started");

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**STEP 5-RUN THE FILE**



**HANDS-ON 2: Load SimpleDateFormat Bean from Spring Configuration XML**

**Step 1: Created XML File**

**Step 2: Added displayDate() method**

public void displayDate() {

LOGGER.info("START");

ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");

SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);

try {

Date date = format.parse("31/12/2018");

LOGGER.debug("Parsed Date: {}", date);

} catch (ParseException e) {

LOGGER.error("Error parsing date", e);

}

LOGGER.info("END");

}

STEP 3- RUN   


**HANDS-ON 3: Add Logging to a Spring Boot Application**

**Step 1: Create application.properties**

logging.level.org.springframework=info

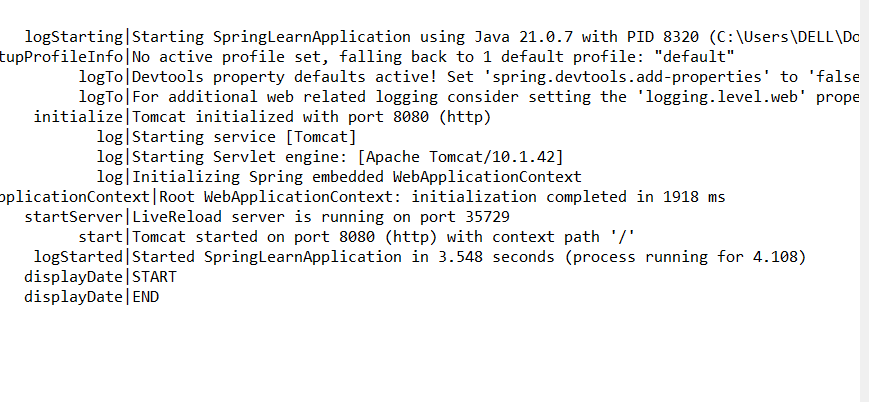
logging.level.com.cognizant.springlearn=debug

logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n

**Step 2: Added Logger to the Class**

private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);

**STEP 3-RUN**



**HANDS-ON 4: Load a Country Bean from Spring XML**

**Step 1: Created Country.java**

package com.cognizant.spring\_learn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

public class Country {

private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);

private String code;

private String name;

public Country() {

LOGGER.debug("Inside Country Constructor");

}

public String getCode() {

LOGGER.debug("Getting code");

return code;

}

public void setCode(String code) {

LOGGER.debug("Setting code");

this.code = code;

}

public String getName() {

LOGGER.debug("Getting name");

return name;

}

public void setName(String name) {

LOGGER.debug("Setting name");

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**Step 2: Created country.xml**

<bean id="country" class="com.cognizant.spring\_learn.Country">

<property name="code" value="IN" />

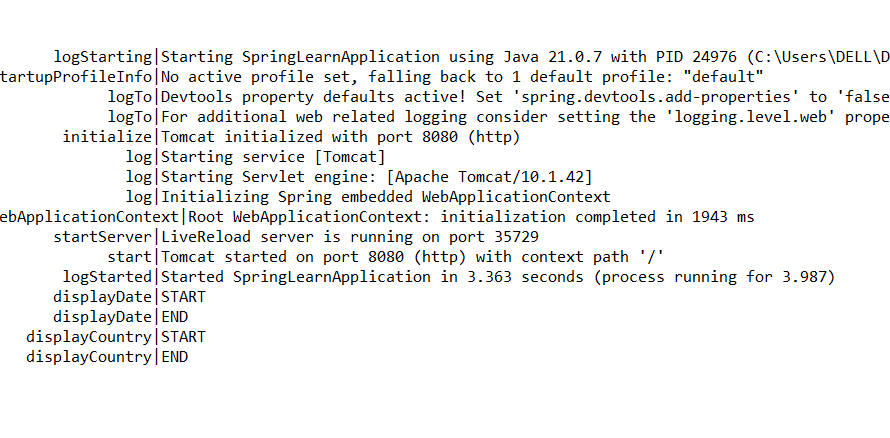
<property name="name" value="India" />

</bean>

**Step 3: Added displayCountry() Method**

ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

Country country = context.getBean("country", Country.class);

LOGGER.debug("Country: {}", country.toString());  
**STEP 4-RUN**  


**HANDS-ON 5: Singleton Scope vs Prototype Scope**

**Step 1: Used Country bean from previous hands-on**

<bean id="country" class="com.cognizant.spring\_learn.Country" scope="singleton">

**Step 2: Called getBean() Twice**

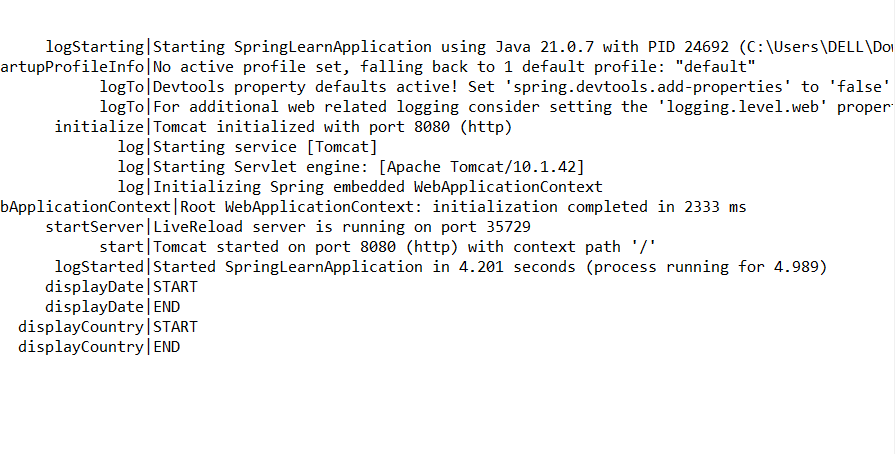
Country country1 = context.getBean("country", Country.class);

Country country2 = context.getBean("country", Country.class);

**Step 3: Updated scope to "prototype"**

<bean id="country" class="com.cognizant.spring\_learn.Country" scope="prototype">

**STEP 4 -RUN**

****

**HANDS-ON 6: Load a List of Country Beans from Spring XML**

**Step 1: Updated country.xml to Include Multiple Beans**

<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>

**Step 2: Added displayCountries() Method**

List<Country> countryList = context.getBean("countryList", List.class);

for (Country c : countryList) {

LOGGER.debug("Country: {}", c);

}

**Step 3: Ran the App**

