Converters and Populators Exercise

1. Work under TelcotraininigCore
2. Create a new Telcotrainingcore-beans.xml and add the POJO definitions for TelcoCustomerData and TelcoRegionData

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
 [y] hybris Platform

 Copyright (c) 2000-2015 hybris AG
 All rights reserved.

 This software is the confidential and proprietary information of hybris
 ("Confidential Information"). You shall not disclose such Confidential
 Information and shall use it only in accordance with the terms of the
 license agreement you entered into with hybris.
-->
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="beans.xsd">

    <bean class="com.virtusa.telco.core.data.TelcoCustomerData">
        <property name="regions" type="java.util.List&lt;com.virtusa.telco.core.data.TelcoRegionData>"/>
        <property name="email" type="java.lang.String"/>
        <property name="name" type="java.lang.String"/>
        <property name="age" type="java.lang.Integer"/>
    </bean>

    <bean class="com.virtusa.telco.core.data.TelcoRegionData">
        <property name="code" type="String"/>
        <property name="name" type="String"/>
    </bean>
</beans>
```

3. Do ant all
4. Now create a populator to populate TelcoCustomerData from the TelcoCustomerModel.

```
public class TelcoCustomerPopulator implements
Populator<TelcoCustomerModel, TelcoCustomerData> {
    @Override
    public void populate(TelcoCustomerModel
telcoCustomerModel, TelcoCustomerData telcoCustomerData)
throws ConversionException {

        Assert.notNull(telcoCustomerModel, "Parameter
telcoCustomerModel cannot be null.");
        Assert.notNull(telcoCustomerData, "Parameter
telcoCustomerData cannot be null.");

        List<TelcoRegionData> regions = new ArrayList<>();

        telcoCustomerModel.getRegion().forEach(telcoRegion ->
        {TelcoRegionData regionData = new TelcoRegionData();
            regionData.setCode(telcoRegion.getCode());
            regionData.setName(telcoRegion.getCode());
            regions.add(regionData);});



        telcoCustomerData.setAge(telcoCustomerModel.getAge());

telcoCustomerData.setEmail(telcoCustomerModel.getUid());

telcoCustomerData.setName(telcoCustomerModel.getName());
        telcoCustomerData.setRegions(regions);
    }
}
```

5. Define a bean for the new populator in the Telcotrainingcore-spring.xml

```
<bean id="telcoCustomerPopulator"

class="com.vertusa.telco.core.populators.TelcoCustomerPopulato
r"/>
```

6. Define a new converter for TelcoCustomerData using this new populator define din step 5

```
<bean id="telcoCustomerConverter"
parent="abstractPopulatingConverter">
    <property name="targetClass"
value="com.virtusa.telco.core.data.TelcoCustomerData"/>
    <property name="populators">
        <list>
            <ref bean="telcoCustomerPopulator"/>
        </list>
    </property>
</bean>
```

7. Inject this new Converter in the EventListener and get the TelcoCustomerData

```
public class NonAdhaarCardCustEventListener extends
AbstractEventListener<NonAdharCardCustomerEvent> {

    @Resource
    Converter<TelcoCustomerModel,TelcoCustomerData>
telcoCustomerConverter;
    @Override
    protected void onEvent(NonAdharCardCustomerEvent
nonAdharCardCustomerEvent) {


nonAdharCardCustomerEvent.getCustomers().forEach(customer ->

System.out.println(telcoCustomerConverter.convert(customer).ge
tName()));

    }
}
```

8. Test the cronjob to see that the logs print the name of the customer.