

# Flexible Search and EventListener Exercise

All changes are to be done in Telcotrainingcore

1. Write a cronjob which gets all the customers which don't have any addhar card number.

```
public class CustomerAddharCardNotificationJob extends
AbstractJobPerformable<CronJobModel> {

    @Autowired
    private CustomerService telcoCustomerService;

    @Autowired
    private EventService eventService;

    @Override
    public PerformResult perform(CronJobModel cronJobModel) {

        Set<TelcoCustomerModel> telcoCustomers =
telcoCustomerService.getAllNonAddharCardCustomers();

        NonAdharCardCustomerEvent customerEvent = new
NonAdharCardCustomerEvent(telcoCustomers);

        eventService.publishEvent(customerEvent);

        return new PerformResult(CronJobResult.SUCCESS,
CronJobStatus.FINISHED);
    }
}
```

Register a bean for the same job in Telcotrainingcore-spring.xml

```
<bean id="telcotrainingCustomerJob"
class="com.vertusa.telco.core.jobs.CustomerAddharCardNotificat
ionJob"
parent="abstractJobPerformable"/>
```

1.a. create a DAO to return a list of customers using flexiblesearchservice.

```
public interface TelcoCustomerDao {
```

```

        public TelcoCustomerModel
getCustomerByAdhharCardNumber(String adhharNo);

        public List<TelcoCustomerModel>
getAllNonAddharCardCustomers();
}

```

#### Implementation of DAO

```

@Service(value = "telcoCustomerDao")
public class DefaultTelcoCustomerDao implements
TelcoCustomerDao {

    @Autowired
    FlexibleSearchService flexibleSearchService;

    private static final String NON_ADHAR_CUSTOMERS= new
StringBuilder().append("SELECT {PK} from ")
                .append("{ "+TelcoCustomerModel._TYPECODE+ " as
TC }")
                .append(" where {TC." +TelcoCustomerModel.ADANO+"}
is null").toString();

    @Override
    public TelcoCustomerModel
getCustomerByAdhharCardNumber(String adhharNo) {
        return null;
    }

    @Override
    public List<TelcoCustomerModel>
getAllNonAddharCardCustomers() {
        FlexibleSearchQuery adahaarCardQuery = new
FlexibleSearchQuery(NON_ADHAR_CUSTOMERS);

        //
adahaarCardQuery.setResultClassList( List<TelcoCustomerModel>)
;

        SearchResult<TelcoCustomerModel> searchResults =
flexibleSearchService.search(adahaarCardQuery);

        if(searchResults.getCount() > 0)
        {
            return searchResults.getResult();
        }
        else
        {
            return Collections.EMPTY_LIST;
        }
    }
}

```

```

    }
}

```

1.b create a service which calls this DAO and converts the returned List of customers to a set.

```

public interface CustomerService {

    public TelcoCustomerModel
getCustomerByAdhharCardNumber(String adhharNo);

    public Set<TelcoCustomerModel>
getAllNonAddharCardCustomers();
}

```

Implementation of Service

```

@Service(value = "telcoCustomerService")
public class DefaultTelcoCustomerService implements
CustomerService {

    public void setTelcoCustomerDao(TelcoCustomerDao
telcoCustomerDao) {
        this.telcoCustomerDao = telcoCustomerDao;
    }

    @Autowired
    private TelcoCustomerDao telcoCustomerDao;

    @Override
    public TelcoCustomerModel
getCustomerByAdhharCardNumber(String adhharNo) {
        return null;
    }

    @Override
    public Set<TelcoCustomerModel>
getAllNonAddharCardCustomers() {
        return new
HashSet<TelcoCustomerModel>(telcoCustomerDao.getAllNonAddharCa
rdCustomers());
    }
}

```

2. Create an event and event listener to work on these customers to log the customer UID into logs.

```

public class NonAdhharCardCustomerEvent extends AbstractEvent {

```

```

private BaseStoreModel baseStore;

private Set<TelcoCustomerModel> customers;
private LanguageModel language;

public BaseStoreModel getBaseStore() {
    return baseStore;
}

public void setBaseStore(BaseStoreModel baseStore) {
    this.baseStore = baseStore;
}

public Set<TelcoCustomerModel> getCustomers() {
    return customers;
}

public void setCustomer(Set<TelcoCustomerModel> customers)
{
    this.customers = customers;
}

public LanguageModel getLanguage() {
    return language;
}

public void setLanguage(LanguageModel language) {
    this.language = language;
}

public NonAdharCardCustomerEvent(Set<TelcoCustomerModel>
customerModels)
{
    this.customers = customerModels;
}
}

```

EventListener

```

public class NonAdhaarCardCustEventListener extends
AbstractEventListener<NonAdharCardCustomerEvent> {

    @Override
    protected void onEvent(NonAdharCardCustomerEvent
nonAdharCardCustomerEvent) {

nonAdharCardCustomerEvent.getCustomers().forEach(customer ->

```

```

System.out.println(customer.getUid()));
    }
}

```

register a bean in Telcotrainingcore-spring.xml

```

<bean id="nonAdhaarCardCustEventListener"
class="com.vertusa.telco.core.event.NonAdhaarCardCustEventList
ener"
parent="abstractEventListener"/>

```

3. Create a JUNIT of Service written in step 1.b using mockito to mock the DAO and testing for the given methods.

```

import de.hybris.bootstrap.annotations.UnitTest;
import junit.framework.Assert;
import org.junit.Before;
import org.junit.Test;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;

import static org.mockito.Mockito.mock;
import static org.mockito.BDDMockito.given;

import static org.junit.Assert.*;

/**
 * Created by abhayphougat on 09/11/15.
 */
@UnitTest
public class DefaultTelcoCustomerServiceTest {

    @Mock
    private DefaultTelcoCustomerDao defaultTelcoCustomerDao;
    private DefaultTelcoCustomerService
defaultTelcoCustomerService;

    @Before
    public void setUp() throws Exception {
        MockitoAnnotations.initMocks(this);
        defaultTelcoCustomerService = new
DefaultTelcoCustomerService();

defaultTelcoCustomerService.setTelcoCustomerDao(defaultTelcoCu
stomerDao);

```

```

    }

    @Test
    public void testGetCustomerByAdhharCardNumber() throws
Exception {

    }

    @Test
    public void testGetAllNonAddharCardCustomers() throws
Exception {
        TelcoCustomerModel customer =
mock(TelcoCustomerModel.class);
        customer.setUid("abhaykumar@gmail.com");
        List<TelcoCustomerModel> returnedList = new
ArrayList<TelcoCustomerModel>();
        returnedList.add(customer);

given(defaultTelcoCustomerDao.getAllNonAddharCardCustomers()).
willReturn(returnedList);

        final Set<TelcoCustomerModel> result =
defaultTelcoCustomerService.getAllNonAddharCardCustomers();

Assert.assertEquals(result.iterator().next(), customer);
    }
}

```

Write an Impex file to create an entry for the cronjob call it  
essentaildata\_telcotrainingcore\_cronjobs.impex

```

INSERT_UPDATE ServicelayerJob;code[unique=true];springId
;telcotrainingCustomerJob;telcotrainingCustomerJob

INSERT_UPDATE CronJob;
code[unique=true];job(code);singleExecutable;sessionLanguage(i
socode)
;telcotrainingCustomerJob;telcotrainingCustomerJob;false;en
INSERT_UPDATE Trigger;cronjob(code)
[unique=true];cronExpression
;telcotrainingCustomerJob; 0 0 0 * * ?

```