***CONTROLLER***

***TRAVELCARECONTROLLER***

package com.kgisl.nsure.travelcare.controller;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.servlet.ModelAndView;

import com.google.gson.Gson;

import com.google.gson.JsonArray;

import com.google.gson.JsonElement;

import com.google.gson.reflect.TypeToken;

import com.kgisl.nsure.authenticate.domain.LoginDO;

import com.kgisl.nsure.authenticate.services.AuthenticateService;

import com.kgisl.nsure.commonutils.domain.CommonutilsDO;

import com.kgisl.nsure.travelcare.domain.TravelCareDO;

import com.kgisl.nsure.travelcare.service.TravelCareService;

@RestController

public class TravelCareController {

@Autowired

private AuthenticateService authService;

@Autowired

TravelCareService travelcareService;

@RequestMapping(value = "/quot", method = RequestMethod.GET)

public ModelAndView quot(@ModelAttribute("login") TravelCareDO traveldo) {

// request.getSession().setAttribute("SESSION\_COUNT",

// request.getSession().getId());

return new ModelAndView("test/quot");

}

@RequestMapping(value = "/client", method = RequestMethod.GET)

public ModelAndView client(@ModelAttribute("login") TravelCareDO traveldo) {

return new ModelAndView("quot/client");

}

// Save

@RequestMapping(value = "/save", method = RequestMethod.GET)

public ModelAndView save(@ModelAttribute("login") TravelCareDO traveldo) {

return new ModelAndView("/save");

}

@RequestMapping(value = "/insert", method = RequestMethod.POST)

public ModelAndView addCustomer(

@ModelAttribute("userFormData") TravelCareDO travelDO,

BindingResult result, HttpServletRequest request,

HttpSession session) {

travelcareService.insert(travelDO);

// System.out.println(userFormData.getName());

return new ModelAndView("quot/client");

}

// UI Grid

@RequestMapping(value = "/fire", method = RequestMethod.GET)

public ModelAndView fire(@ModelAttribute("login") TravelCareDO traveldo) {

return new ModelAndView("fire");

}

// UI Grid

@RequestMapping(value = "/covernoteGrid", method = RequestMethod.GET)

public @ResponseBody

String covernoteGrid(TravelCareDO quotationDO, HttpServletRequest request) {

List<TravelCareDO> covernotegrid = null;

ArrayList<TravelCareDO> mainList = new ArrayList<TravelCareDO>();

System.out.println("covernoteGrid");

String jsonString = null;

Gson gson = new Gson();

try {

covernotegrid = travelcareService.getcnGrid(quotationDO);

mainList.addAll(0, covernotegrid);

jsonString = gson.toJson(mainList);

System.out.println("covernotegrid\n" + jsonString);

} catch (Exception e) {

System.out.println(e);

}

return jsonString;

}

// DropDowns

@RequestMapping(value = "/testselect", method = RequestMethod.POST)

public @ResponseBody

String getEndorCategory(HttpServletRequest request) {

ArrayList<TravelCareDO> mainList = new ArrayList<TravelCareDO>();

List<TravelCareDO> endorCategoryList = null;

List<TravelCareDO> genderList = null;

List<TravelCareDO> MList = null;

List<TravelCareDO> SourceList = null;

List<TravelCareDO> BankBranchList = null;

List<TravelCareDO> NationalityList = null;

List<TravelCareDO> RaceList = null;

List<TravelCareDO> TradeList = null;

List<TravelCareDO> StateList = null;

List<TravelCareDO> CountryList = null;

List<TravelCareDO> SalutationList = null;

List<TravelCareDO> IncomeList = null;

String jsonString = null;

try {

// System.out.println("33333\n");

endorCategoryList = travelcareService.getEndorCategoryList(null);

genderList = travelcareService.getGenderList(null);

MList = travelcareService.getMList(null);

SourceList = travelcareService.getSourceList(null);

BankBranchList = travelcareService.getBankBranchList(null);

NationalityList = travelcareService.getNationalityList(null);

RaceList = travelcareService.getRaceList(null);

TradeList = travelcareService.getTradeList(null);

StateList = travelcareService.getStateList(null);

CountryList = travelcareService.getCountryList(null);

SalutationList = travelcareService.getSalutationList(null);

IncomeList = travelcareService.getIncomeList(null);

mainList.addAll(0, SourceList);

mainList.addAll(1, BankBranchList);

mainList.addAll(2, endorCategoryList);

mainList.addAll(3, NationalityList);

mainList.addAll(4, RaceList);

mainList.addAll(5, genderList);

mainList.addAll(6, MList);

mainList.addAll(7, TradeList);

mainList.addAll(8, StateList);

mainList.addAll(9, CountryList);

mainList.addAll(10, SalutationList);

mainList.addAll(11, IncomeList);

// System.out.println("start");

Gson gson = new Gson();

jsonString = gson.toJson(mainList);

// System.out.println("dummy\n" + jsonString);

// System.out.println("end");

} catch (Exception e) {

System.out.println(e);

}

return jsonString;

}

// quotation(test)

@RequestMapping(value = "/quotation", method = RequestMethod.GET)

public ModelAndView CRCInitialPage(

@ModelAttribute("loginDO") LoginDO login, HttpServletRequest request)

throws Exception {

request.getSession().setAttribute("SESSION\_COUNT",

request.getSession().getId());

int j = travelcareService.checkUserName("mythily");

System.out.println(j + " GOOD\n");

// return new ModelAndView("auth/crclogin");

return new ModelAndView("/quot/quotation");

}

}

***DAO***

***TRAVELCAREDAO***

**package** com.kgisl.nsure.travelcare.dao;

**import** java.util.List;

**import** com.kgisl.nsure.commonutils.domain.CommonutilsDO;

**import** com.kgisl.nsure.travelcare.domain.TravelCareDO;

**public** **interface** TravelCareDAO {

**public** **void** GetDatabaseTable(TravelCareDO travelDO);

**int** GetDatabaseTable(String userName);

**public** **void** CheckUserName(String userName);

// DropDowns

**public** List<TravelCareDO> getEndorCategoryList(TravelCareDO travelDO);

**public** List<TravelCareDO> getGenderList(TravelCareDO travelDO);

**public** List<TravelCareDO> getMList(TravelCareDO travelDO);

**public** List<TravelCareDO> getSourceList(TravelCareDO travelDO);

**public** List<TravelCareDO> getBankBranchList(TravelCareDO travelDO);

**public** List<TravelCareDO> getNationalityList(TravelCareDO travelDO);

**public** List<TravelCareDO> getRaceList(TravelCareDO travelDO);

**public** List<TravelCareDO> getTradeList(TravelCareDO travelDO);

**public** List<TravelCareDO> getStateList(TravelCareDO travelDO);

**public** List<TravelCareDO> getCountryList(TravelCareDO travelDO);

**public** List<TravelCareDO> getSalutationList(TravelCareDO travelDO);

**public** List<TravelCareDO> getIncomeList(TravelCareDO travelDO);

// UI Grid

List<TravelCareDO> covernoteGrid(TravelCareDO quotationDO);

// Save

**public** **void** insert(TravelCareDO quotationDO);

}

***TRAVELCAREDAOIMPL***

**package** com.kgisl.nsure.travelcare.dao;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Map;

**import** javax.sql.DataSource;

**import** oracle.jdbc.OracleTypes;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.jdbc.core.JdbcTemplate;

**import** org.springframework.jdbc.core.RowMapper;

**import** org.springframework.jdbc.core.SqlParameter;

**import** org.springframework.jdbc.core.namedparam.MapSqlParameterSource;

**import** org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;

**import** org.springframework.jdbc.core.namedparam.SqlParameterSource;

**import** org.springframework.jdbc.core.simple.SimpleJdbcCall;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.kgisl.nsure.travelcare.domain.TravelCareDO;

@Service

@Transactional

**public** **class** TravelCareDAOImpl **implements** TravelCareDAO {

@Autowired

DataSource dataSource;

**private** JdbcTemplate jdbcTemplate;

**private** NamedParameterJdbcTemplate namedParameterJdbcTemplate;

@Autowired

**public** TravelCareDAOImpl(DataSource dataSource) {

jdbcTemplate = **new** JdbcTemplate(dataSource);

namedParameterJdbcTemplate = **new** NamedParameterJdbcTemplate(dataSource);

}

// test

@Override

**public** List<TravelCareDO> getEndorCategoryList(TravelCareDO travelDO) {

List<TravelCareDO> endorCategoryList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT ID, NAME FROM Type\_ID";

endorCategoryList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setNUM\_ID(rs.getInt("ID"));

commonutilsDO.setNAME(rs.getString("NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** endorCategoryList;

}

// Gender

@Override

**public** List<TravelCareDO> getGenderList(TravelCareDO travelDO) {

List<TravelCareDO> genderList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_GENDER\_ID, VCH\_GENDER\_NAME FROM EIS\_MST\_OTS\_GENDER";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

genderList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setGEN\_ID(rs.getInt("NUM\_GENDER\_ID"));

commonutilsDO.setGENDER(rs

.getString("VCH\_GENDER\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** genderList;

}

// Marital Status

@Override

**public** List<TravelCareDO> getMList(TravelCareDO travelDO) {

List<TravelCareDO> MList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_MARITAL\_STATUS\_ID, VCH\_MARITAL\_STATUS\_NAME FROM EIS\_MST\_OTS\_MARITAL\_STATUS";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

MList = jdbcTemplate.query(finalSQL, **new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setM\_ID(rs.getInt("NUM\_MARITAL\_STATUS\_ID"));

commonutilsDO.setSTATE(rs

.getString("VCH\_MARITAL\_STATUS\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** MList;

}

// Source Bank

@Override

**public** List<TravelCareDO> getSourceList(TravelCareDO travelDO) {

List<TravelCareDO> SourceList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_SOURCE\_TYPE, VCH\_SOURCE\_TYPE\_NAME FROM EIS\_MST\_OTS\_SOURCE\_TYPE";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

SourceList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setSOURCE\_ID(rs

.getInt("NUM\_SOURCE\_TYPE"));

commonutilsDO.setSOURCE\_TYPE(rs

.getString("VCH\_SOURCE\_TYPE\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** SourceList;

}

// Bank Branch

@Override

**public** List<TravelCareDO> getBankBranchList(TravelCareDO travelDO) {

List<TravelCareDO> BankBranchList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_SOURCE\_BANK\_BRANCH\_ID, VCH\_SOURCE\_BANK\_BRANCH\_NAME FROM EIS\_MST\_OTS\_SOURCE\_BANK\_BRANCH";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

BankBranchList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setSOURCE\_BANK\_ID(rs

.getInt("NUM\_SOURCE\_BANK\_BRANCH\_ID"));

commonutilsDO.setSOURCE\_BANK\_NAME(rs

.getString("VCH\_SOURCE\_BANK\_BRANCH\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** BankBranchList;

}

// Nationality

@Override

**public** List<TravelCareDO> getNationalityList(TravelCareDO travelDO) {

List<TravelCareDO> NationalityList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_NATIONALITY\_ID, VCH\_NATIONALITY\_NAME FROM EIS\_MST\_OTS\_NATIONALITY";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

NationalityList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setNATIONALITY\_ID(rs

.getInt("NUM\_NATIONALITY\_ID"));

commonutilsDO.setNATIONALITY\_NAME(rs

.getString("VCH\_NATIONALITY\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** NationalityList;

}

// Race

@Override

**public** List<TravelCareDO> getRaceList(TravelCareDO travelDO) {

List<TravelCareDO> RaceList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_RACE\_ID, VCH\_RACE\_NAME FROM EIS\_MST\_OTS\_RACE";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

RaceList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setRACE\_ID(rs.getInt("NUM\_RACE\_ID"));

commonutilsDO.setRACE\_NAME(rs

.getString("VCH\_RACE\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** RaceList;

}

// Trade

@Override

**public** List<TravelCareDO> getTradeList(TravelCareDO travelDO) {

List<TravelCareDO> TradeList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_TRADE\_ID, VCH\_TRADE\_NAME FROM EIS\_MST\_OTS\_TRADE";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

TradeList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setTRADE\_ID(rs.getInt("NUM\_TRADE\_ID"));

commonutilsDO.setTRADE\_NAME(rs

.getString("VCH\_TRADE\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** TradeList;

}

// State

@Override

**public** List<TravelCareDO> getStateList(TravelCareDO travelDO) {

List<TravelCareDO> StateList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_STATE\_ID, VCH\_STATE\_NAME FROM EIS\_MST\_OTS\_STATE";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

StateList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setSTATE\_ID(rs.getInt("NUM\_STATE\_ID"));

commonutilsDO.setSTATE\_NAME(rs

.getString("VCH\_STATE\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** StateList;

}

// Country

@Override

**public** List<TravelCareDO> getCountryList(TravelCareDO travelDO) {

List<TravelCareDO> CountryList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_COUNTRY\_ID, VCH\_COUNTRY\_NAME FROM EIS\_MST\_OTS\_COUNTRY";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

CountryList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setCOUNTRY\_ID(rs

.getInt("NUM\_COUNTRY\_ID"));

commonutilsDO.setCOUNTRY\_NAME(rs

.getString("VCH\_COUNTRY\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** CountryList;

}

// Salutation

@Override

**public** List<TravelCareDO> getSalutationList(TravelCareDO travelDO) {

List<TravelCareDO> SalutationList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_SALUTATION\_ID, VCH\_SALUTATION\_NAME FROM EIS\_MST\_OTS\_SALUTATION";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

SalutationList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setSALUTATION\_ID(rs

.getInt("NUM\_SALUTATION\_ID"));

commonutilsDO.setSALUTATION\_NAME(rs

.getString("VCH\_SALUTATION\_NAME"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** SalutationList;

}

// Income

@Override

**public** List<TravelCareDO> getIncomeList(TravelCareDO travelDO) {

List<TravelCareDO> IncomeList = **null**;

**try** {

String finalSQL = **null**;

finalSQL = "SELECT NUM\_INCOME\_ID, NUM\_INCOME\_AMOUNT FROM EIS\_MST\_OTS\_INCOME\_RANGE";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(finalSQL);

System.*out*.println(customers + "Mythily");

IncomeList = jdbcTemplate.query(finalSQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO commonutilsDO = **new** TravelCareDO();

commonutilsDO.setINCOME\_ID(rs

.getInt("NUM\_INCOME\_ID"));

commonutilsDO.setINCOME\_AMOUNT(rs

.getString("NUM\_INCOME\_AMOUNT"));

**return** commonutilsDO;

}

});

} **catch** (Exception e) {

e.printStackTrace();

}

**return** IncomeList;

}

// UI Grid

@Override

**public** List<TravelCareDO> covernoteGrid(TravelCareDO quotationDO) {

List<TravelCareDO> covernoteGrid = **null**;

**try** {

String SQL = **null**;

SQL = "SELECT \* FROM EIS\_MST\_OTS\_INTEREST\_CALC";

covernoteGrid = jdbcTemplate.query(SQL,

**new** RowMapper<TravelCareDO>() {

@Override

**public** TravelCareDO mapRow(ResultSet rs, **int** rowNum)

**throws** SQLException {

TravelCareDO quotationDO = **new** TravelCareDO();

quotationDO.setNUM(rs.getInt("NUM"));

quotationDO.setINTEREST\_CODE(rs

.getString("INTEREST\_CODE"));

quotationDO.setINTEREST\_DESC(rs

.getString("INTEREST\_DESC"));

quotationDO.setSUM\_INSURED(rs

.getString("SUM\_INSURED"));

quotationDO.setRATE(rs.getString("RATE"));

quotationDO.setPREMIUM(rs.getString("PREMIUM"));

**return** quotationDO;

}

});

} **catch** (Exception e) {

System.*out*.println(e);

}

**return** covernoteGrid;

}

/\*

\* //Source@Override public List<TravelCareDO> getSourceType(TravelCareDO

\* travelDO) { List<TravelCareDO> SourceType = null; try{ String finalSQL =

\* null; finalSQL

\* ="SELECT NUM\_SOURCE\_TYPE, VCH\_SOURCE\_TYPE\_NAME FROM EIS\_MST\_OTS\_SOURCE\_TYPE"

\* ; List<Map<String, Object>> customers =

\* jdbcTemplate.queryForList(finalSQL); System.out.println(customers +

\* "Mythily");

\*

\* SourceType = jdbcTemplate.query(finalSQL,new RowMapper<TravelCareDO>() {

\*

\* @Override public TravelCareDO mapRow(ResultSet rs, int rowNum) throws

\* SQLException { TravelCareDO commonutilsDO = new TravelCareDO();

\* commonutilsDO.setSOURCE\_TYPE\_ID(rs.getInt("NUM\_SOURCE\_TYPE"));

\* commonutilsDO.setSOURCE\_TYPE\_NAME(rs.getString("VCH\_SOURCE\_TYPE\_NAME"));

\* return commonutilsDO; } }); }catch(Exception e){ e.printStackTrace(); }

\* return SourceType; }

\*/

// Save

@Override

**public** **void** insert(TravelCareDO quotationDO) {

// **TODO** Auto-generated method stub

// System.out.println(quotationDO.getName()+" mohn");

// System.out.println(" 11111"+quotationDO.getContactName());

// quotationDO.getContactName();

**try** {

// System.out.println(quotationDO.getName() + "1222");

System.*out*.println(quotationDO.getName());

SimpleJdbcCall simpleJdbcCall = **new** SimpleJdbcCall(dataSource);

simpleJdbcCall

.withCatalogName("PKG\_FIRE\_SAVE")

.withProcedureName("PRQUOTATIONSAVE")

.withoutProcedureColumnMetaDataAccess()

.declareParameters(

**new** SqlParameter("P\_NUM\_ID", OracleTypes.*NUMBER*),

**new** SqlParameter("P\_VCH\_NAME", OracleTypes.*VARCHAR*));

SqlParameterSource inputParams = **new** MapSqlParameterSource()

.addValue("P\_NUM\_ID", quotationDO.getPol\_ID()).addValue(

"P\_VCH\_NAME", quotationDO.getName());

// System.out.println(quotationDO.getName()+" 11111");

Map<String, Object> transactionStatus = simpleJdbcCall

.execute(inputParams);

} **catch** (Exception e) {

System.*out*.println();

}

System.*out*.println("DAO<");

}

@Override

**public** **int** GetDatabaseTable(String userName) {

Integer num;

num = **new** Integer(10);

**try** {

System.*out*.println("\*\*\*\*test init\*\*\*\*");

System.*out*.println("AAA");

String sql = "SELECT \* FROM ETS\_MTS\_OTS";

List<Map<String, Object>> customers = jdbcTemplate

.queryForList(sql);

System.*out*.println(customers + "Mythily");

} **catch** (Exception e) {

System.*out*.println(e + "AA");

}

**return** num;

}

@Override

**public** **void** GetDatabaseTable(TravelCareDO travelcareDO) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** CheckUserName(String userName) {

// **TODO** Auto-generated method stub

}

}

***SERVICE***

***TRAVELCARESERVICE***

**package** com.kgisl.nsure.travelcare.service;

**import** java.util.List;

**import** com.kgisl.nsure.commonutils.domain.CommonutilsDO;

**import** com.kgisl.nsure.travelcare.domain.TravelCareDO;

**public** **interface** TravelCareService {

**int** checkUserName(String userName);

**public** List<TravelCareDO> getEndorCategoryList(TravelCareDO travelDO);

**public** List<TravelCareDO> getGenderList(TravelCareDO travelDO);

**public** List<TravelCareDO> getMList(TravelCareDO travelDO);

**public** List<TravelCareDO> getSourceList(TravelCareDO travelDO);

**public** List<TravelCareDO> getBankBranchList(TravelCareDO travelDO);

**public** List<TravelCareDO> getNationalityList(TravelCareDO travelDO);

**public** List<TravelCareDO> getRaceList(TravelCareDO travelDO);

**public** List<TravelCareDO> getTradeList(TravelCareDO travelDO);

**public** List<TravelCareDO> getStateList(TravelCareDO travelDO);

**public** List<TravelCareDO> getCountryList(TravelCareDO travelDO);

**public** List<TravelCareDO> getSalutationList(TravelCareDO travelDO);

**public** List<TravelCareDO> getIncomeList(TravelCareDO travelDO);

List<TravelCareDO> getcnGrid(TravelCareDO quotationDO);

// public List<TravelCareDO> getSourceType(TravelCareDO travelDO);

**void** insert(TravelCareDO travelDO);

}

***TRAVELCARESERVICEIMPL***

**package** com.kgisl.nsure.travelcare.service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.kgisl.nsure.commonutils.domain.CommonutilsDO;

**import** com.kgisl.nsure.travelcare.dao.TravelCareDAO;

**import** com.kgisl.nsure.travelcare.domain.TravelCareDO;

@Service

**public** **class** TravelCareServiceImpl **implements** TravelCareService {

@Autowired

TravelCareDAO travelcaredao;

@Override

**public** List<TravelCareDO> getEndorCategoryList(TravelCareDO travelDO) {

List<TravelCareDO> endorCategoryList = **null**;

**try** {

endorCategoryList = travelcaredao.getEndorCategoryList(**null**);

// System.out.println("0000"+endorCategoryList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** endorCategoryList;

}

@Override

**public** List<TravelCareDO> getGenderList(TravelCareDO travelDO) {

List<TravelCareDO> genderList = **null**;

**try** {

genderList = travelcaredao.getGenderList(**null**);

// System.out.println("1111"+genderList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** genderList;

}

@Override

**public** List<TravelCareDO> getMList(TravelCareDO travelDO) {

List<TravelCareDO> MList = **null**;

**try** {

MList = travelcaredao.getMList(**null**);

// System.out.println("2222"+MList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** MList;

}

@Override

**public** List<TravelCareDO> getSourceList(TravelCareDO travelDO) {

List<TravelCareDO> SourceList = **null**;

**try** {

SourceList = travelcaredao.getSourceList(**null**);

// System.out.println("1111"+SourceList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** SourceList;

}

@Override

**public** List<TravelCareDO> getBankBranchList(TravelCareDO travelDO) {

List<TravelCareDO> BankBranchList = **null**;

**try** {

BankBranchList = travelcaredao.getBankBranchList(**null**);

// System.out.println("2222"+BankBranchList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** BankBranchList;

}

@Override

**public** List<TravelCareDO> getNationalityList(TravelCareDO travelDO) {

List<TravelCareDO> NationalityList = **null**;

**try** {

NationalityList = travelcaredao.getNationalityList(**null**);

// System.out.println("3333"+NationalityList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** NationalityList;

}

@Override

**public** List<TravelCareDO> getRaceList(TravelCareDO travelDO) {

List<TravelCareDO> RaceList = **null**;

**try** {

RaceList = travelcaredao.getRaceList(**null**);

// System.out.println("3333"+RaceList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** RaceList;

}

@Override

**public** List<TravelCareDO> getTradeList(TravelCareDO travelDO) {

List<TravelCareDO> TradeList = **null**;

**try** {

TradeList = travelcaredao.getTradeList(**null**);

// System.out.println("3333"+TradeList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** TradeList;

}

@Override

**public** List<TravelCareDO> getStateList(TravelCareDO travelDO) {

List<TravelCareDO> StateList = **null**;

**try** {

StateList = travelcaredao.getStateList(**null**);

// System.out.println("3333"+StateList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** StateList;

}

@Override

**public** List<TravelCareDO> getCountryList(TravelCareDO travelDO) {

List<TravelCareDO> CountryList = **null**;

**try** {

CountryList = travelcaredao.getCountryList(**null**);

// System.out.println("3333"+CountryList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** CountryList;

}

@Override

**public** List<TravelCareDO> getSalutationList(TravelCareDO travelDO) {

List<TravelCareDO> SalutationList = **null**;

**try** {

SalutationList = travelcaredao.getSalutationList(**null**);

// System.out.println("3333"+SalutationList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** SalutationList;

}

@Override

**public** List<TravelCareDO> getIncomeList(TravelCareDO travelDO) {

List<TravelCareDO> IncomeList = **null**;

**try** {

IncomeList = travelcaredao.getIncomeList(**null**);

// System.out.println("3333"+IncomeList);

} **catch** (Exception e) {

// throw new Exception(e);

e.printStackTrace();

}

**return** IncomeList;

}

// UI Grid

@Override

**public** List<TravelCareDO> getcnGrid(TravelCareDO quotationDO) {

List<TravelCareDO> covernoteGrid = **null**;

**try** {

covernoteGrid = travelcaredao.covernoteGrid(quotationDO);

} **catch** (Exception e) {

System.*out*.println(e);

}

**return** covernoteGrid;

}

// Save

@Override

**public** **void** insert(TravelCareDO quotationDO) {

// **TODO** Auto-generated method stub

System.*out*.println("Serv>");

**if** (quotationDO.getName() != **null**)

System.*out*.println(quotationDO.getName().toString());

travelcaredao.insert(quotationDO);

System.*out*.println("Serv>");

}

/\*

\* @Override public List<TravelCareDO> getSourceType(TravelCareDO travelDO)

\* { List<TravelCareDO> SourceType = null; try{ SourceType =

\* travelcaredao.getSourceType(null);

\* //System.out.println("3333"+IncomeList); }catch(Exception e){ //throw new

\* Exception(e); e.printStackTrace(); } return SourceType; }

\*/

@Override

**public** **int** checkUserName(String userName) {

System.*out*.println("mythily");

**return** travelcaredao.GetDatabaseTable(userName);

}

}

***DOMAIN***

***TRAVELCAREDO***

package com.kgisl.nsure.travelcare.domain;

import java.util.Date;

import org.springframework.stereotype.Component;

@Component

public class TravelCareDO {

private int NUM\_ID;

private String NAME;

private int GEN\_ID;

private String GENDER;

private int M\_ID;

private String STATE;

private int SOURCE\_ID;

private String SOURCE\_TYPE;

private int SOURCE\_BANK\_ID;

private String SOURCE\_BANK\_NAME;

private int NATIONALITY\_ID;

private String NATIONALITY\_NAME;

private int RACE\_ID;

private String RACE\_NAME;

private int TRADE\_ID;

private String TRADE\_NAME;

private int STATE\_ID;

private String STATE\_NAME;

private int COUNTRY\_ID;

private String COUNTRY\_NAME;

private int SALUTATION\_ID;

private String SALUTATION\_NAME;

private int INCOME\_ID;

private String INCOME\_AMOUNT;

private int SOURCE\_TYPE\_ID;

private String SOURCE\_TYPE\_NAME;

private int TEST\_ID;

private String TEST\_NAME;

// UI Grid

private int NUM;

private String INTEREST\_CODE;

private String INTEREST\_DESC;

private String SUM\_INSURED;

private String RATE;

private String PREMIUM;

// Save

private int Pol\_ID;

// private String Pol\_Name;

private String name;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getTEST\_ID() {

return TEST\_ID;

}

public void setTEST\_ID(int tEST\_ID) {

TEST\_ID = tEST\_ID;

}

public String getTEST\_NAME() {

return TEST\_NAME;

}

public void setTEST\_NAME(String tEST\_NAME) {

TEST\_NAME = tEST\_NAME;

}

public int getPol\_ID() {

return Pol\_ID;

}

public void setPol\_ID(int pol\_ID) {

Pol\_ID = pol\_ID;

}

// public String getPol\_Name() {

// return Pol\_Name;

// }

public int getNUM() {

return NUM;

}

public void setNUM(int nUM) {

NUM = nUM;

}

public String getINTEREST\_CODE() {

return INTEREST\_CODE;

}

public void setINTEREST\_CODE(String iNTEREST\_CODE) {

INTEREST\_CODE = iNTEREST\_CODE;

}

public String getINTEREST\_DESC() {

return INTEREST\_DESC;

}

public void setINTEREST\_DESC(String iNTEREST\_DESC) {

INTEREST\_DESC = iNTEREST\_DESC;

}

public String getSUM\_INSURED() {

return SUM\_INSURED;

}

public void setSUM\_INSURED(String sUM\_INSURED) {

SUM\_INSURED = sUM\_INSURED;

}

public String getRATE() {

return RATE;

}

public void setRATE(String rATE) {

RATE = rATE;

}

public String getPREMIUM() {

return PREMIUM;

}

public void setPREMIUM(String pREMIUM) {

PREMIUM = pREMIUM;

}

/\*

\* public int getTEST\_ID() { return TEST\_ID; } public void setTEST\_ID(int

\* tYPE\_ID) { TEST\_ID = tYPE\_ID; } public String getTEST\_NAME() { return

\* TEST\_NAME; } public void setTEST\_NAME(String tEST\_NAME) { TEST\_NAME =

\* tEST\_NAME; }

\*/

public TravelCareDO() {

super();

// TODO Auto-generated constructor stub

}

public int getNUM\_ID() {

return NUM\_ID;

}

public void setNUM\_ID(int nUM\_ID) {

NUM\_ID = nUM\_ID;

}

public String getNAME() {

return NAME;

}

public void setNAME(String nAME) {

NAME = nAME;

}

public int getGEN\_ID() {

return GEN\_ID;

}

public void setGEN\_ID(int gEN\_ID) {

GEN\_ID = gEN\_ID;

}

public String getGENDER() {

return GENDER;

}

public void setGENDER(String gENDER) {

GENDER = gENDER;

}

public int getM\_ID() {

return M\_ID;

}

public void setM\_ID(int m\_ID) {

M\_ID = m\_ID;

}

public String getSTATE() {

return STATE;

}

public void setSTATE(String sTATE) {

STATE = sTATE;

}

public int getSOURCE\_ID() {

return SOURCE\_ID;

}

public void setSOURCE\_ID(int sOURCE\_ID) {

SOURCE\_ID = sOURCE\_ID;

}

public String getSOURCE\_TYPE() {

return SOURCE\_TYPE;

}

public void setSOURCE\_TYPE(String sOURCE\_TYPE) {

SOURCE\_TYPE = sOURCE\_TYPE;

}

public int getSOURCE\_BANK\_ID() {

return SOURCE\_BANK\_ID;

}

public void setSOURCE\_BANK\_ID(int sOURCE\_BANK\_ID) {

SOURCE\_BANK\_ID = sOURCE\_BANK\_ID;

}

public String getSOURCE\_BANK\_NAME() {

return SOURCE\_BANK\_NAME;

}

public void setSOURCE\_BANK\_NAME(String sOURCE\_BANK\_NAME) {

SOURCE\_BANK\_NAME = sOURCE\_BANK\_NAME;

}

public int getNATIONALITY\_ID() {

return NATIONALITY\_ID;

}

public void setNATIONALITY\_ID(int nATIONALITY\_ID) {

NATIONALITY\_ID = nATIONALITY\_ID;

}

public String getNATIONALITY\_NAME() {

return NATIONALITY\_NAME;

}

public void setNATIONALITY\_NAME(String nATIONALITY\_NAME) {

NATIONALITY\_NAME = nATIONALITY\_NAME;

}

public int getRACE\_ID() {

return RACE\_ID;

}

public void setRACE\_ID(int rACE\_ID) {

RACE\_ID = rACE\_ID;

}

public String getRACE\_NAME() {

return RACE\_NAME;

}

public void setRACE\_NAME(String rACE\_NAME) {

RACE\_NAME = rACE\_NAME;

}

public int getTRADE\_ID() {

return TRADE\_ID;

}

public void setTRADE\_ID(int tRADE\_ID) {

TRADE\_ID = tRADE\_ID;

}

public String getTRADE\_NAME() {

return TRADE\_NAME;

}

public void setTRADE\_NAME(String tRADE\_NAME) {

TRADE\_NAME = tRADE\_NAME;

}

public int getSTATE\_ID() {

return STATE\_ID;

}

public void setSTATE\_ID(int sTATE\_ID) {

STATE\_ID = sTATE\_ID;

}

public String getSTATE\_NAME() {

return STATE\_NAME;

}

public void setSTATE\_NAME(String sTATE\_NAME) {

STATE\_NAME = sTATE\_NAME;

}

public int getCOUNTRY\_ID() {

return COUNTRY\_ID;

}

public void setCOUNTRY\_ID(int cOUNTRY\_ID) {

COUNTRY\_ID = cOUNTRY\_ID;

}

public String getCOUNTRY\_NAME() {

return COUNTRY\_NAME;

}

public void setCOUNTRY\_NAME(String cOUNTRY\_NAME) {

COUNTRY\_NAME = cOUNTRY\_NAME;

}

public int getSALUTATION\_ID() {

return SALUTATION\_ID;

}

public void setSALUTATION\_ID(int sALUTATION\_ID) {

SALUTATION\_ID = sALUTATION\_ID;

}

public String getSALUTATION\_NAME() {

return SALUTATION\_NAME;

}

public void setSALUTATION\_NAME(String sALUTATION\_NAME) {

SALUTATION\_NAME = sALUTATION\_NAME;

}

public int getINCOME\_ID() {

return INCOME\_ID;

}

public void setINCOME\_ID(int iNCOME\_ID) {

INCOME\_ID = iNCOME\_ID;

}

public String getINCOME\_AMOUNT() {

return INCOME\_AMOUNT;

}

public void setINCOME\_AMOUNT(String iNCOME\_AMOUNT) {

INCOME\_AMOUNT = iNCOME\_AMOUNT;

}

public int getSOURCE\_TYPE\_ID() {

return SOURCE\_TYPE\_ID;

}

public void setSOURCE\_TYPE\_ID(int sOURCE\_TYPE\_ID) {

SOURCE\_TYPE\_ID = sOURCE\_TYPE\_ID;

}

public String getSOURCE\_TYPE\_NAME() {

return SOURCE\_TYPE\_NAME;

}

public void setSOURCE\_TYPE\_NAME(String sOURCE\_TYPE\_NAME) {

SOURCE\_TYPE\_NAME = sOURCE\_TYPE\_NAME;

}

}

***JSP PAGES***

***INDEX.JSP***

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<%

**if**(request.getSession()!=**null** && request.getSession().getAttribute("logouturl")!=**null**){

response.sendRedirect(request.getContextPath()+request.getSession().getAttribute("logouturl"));

}**else**{

response.sendRedirect(request.getContextPath()+"/admin/");

}

%>

***CLIENT.JSP***

<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<%

**if**(request.getSession()!=**null** && request.getSession().getAttribute("logouturl")!=**null**){

response.sendRedirect(request.getContextPath()+request.getSession().getAttribute("logouturl"));

}**else**{

response.sendRedirect(request.getContextPath()+"/admin/");

}

%>

***SAVE.JSP***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<form modelAttribute=*"userFormData"* action=*"insert"* method=*"POST"*>

<input type=*"text"* name=*"Name"* value=*""* />

<button onclick="">Click</button>

</form>

</body>

</html>

***GRID.JSP***

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<meta charset=*"UTF-8"*>

<title>angular-ui-grid enableFullRowSelection</title>

<script src=*'resources/js/angular.min.js'*></script>

<script src=*'resources/js/angular-touch.min.js'*></script>

<script src=*'resources/js/angular-animate.min.js'*></script>

<script src=*'resources/js/ui-grid.min.js'*></script>

<script src=*"resources/js/index.js"*></script>

<script src=*"resources/js/bootstrap-model.js"*></script>

<link rel=*'stylesheet prefetch'* href=*'resources/css/ui-grid.min.css'*>

<link rel=*"stylesheet"*

href=*"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script

src=*"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"*></script>

<script

src=*"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

<script>

**if**(angular.element($("#ENDOR\_NON\_JPJ\_MOTOR")).scope() != undefined){

**var** parentScope = angular.element($("#ENDOR\_NON\_JPJ\_MOTOR")).scope();

**var** sourceTypeGroupId = parentScope.motorclientdet.sourceTypeGroupId;

**if**(parentScope!=**null** && parentScope!=undefined){

parentScope.motorclientdet.GEN\_ID = rowEntity.GEN\_ID;

parentScope.motorclientdet.GENDER = rowEntity.GENDER;

parentScope.motorclientdet.branchName = rowEntity.branchName;

parentScope.motorclientdet.sourceTypeGroupId = sourceTypeGroupId;

}

self.close(**this**);

}

</script>

</head>

<body>

<button type=*"button"* class=*"btn btn-info btn-lg"* data-toggle=*"modal"*

data-target=*"#myModal"*>Grid</button>

<!-- Modal -->

<div class=*"modal fade"* id=*"myModal"* role=*"dialog"*>

<div class=*"modal-dialog"*>

<!-- Modal content-->

<div class=*"modal-content"*>

<div class=*"modal-header"*>

<!-- <button type="button" class="close" data-dismiss="modal">&times;</button> -->

<h4 class=*"modal-title"*></h4>

</div>

<div data-ng-app=*"app"*>

<div data-ng-controller=*"covernoteGrid"*>

<div data-ui-grid=*"gridOptions"* data-ui-grid-selection

class=*"grid"*></div>

</div>

</div>

<div class=*"modal-footer"*>

<button type=*"button"* class=*"btn btn-default"* data-dismiss=*"modal"*>Close</button>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

***POM.XML***

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>RHBI</groupId>

<artifactId>RHBI</artifactId>

<packaging>war</packaging>

<version>1.0</version>

<name>RHB Insurance Application</name>

<url>http://maven.apache.org</url>

<repositories>

<repository>

<id>mavencentral</id>

<name>Maven Central</name>

<url>https://repo1.maven.org/maven2/</url>

<layout>default</layout>

</repository>

<repository>

<id>localrepo</id>

<name>localrepo-releases</name>

<url>http://10.100.7.76:8081/artifactory/ext-release-local/</url>

</repository>

</repositories>

<properties>

<jaxws.scope>compile</jaxws.scope>

<jdk.version>1.7</jdk.version>

<cxf.version>3.0.4</cxf.version>

<spring.version>4.1.6.RELEASE</spring.version>

<spring.security.version>4.0.1.RELEASE</spring.security.version>

<slf4j.version>1.7.12</slf4j.version>

<spring.ws.version>2.2.4.RELEASE</spring.ws.version>

<apache.ws.version>1.6.19</apache.ws.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

</dependency>

<dependency>

<groupId>com.jayway.restassured</groupId>

<artifactId>spring-mock-mvc</artifactId>

<version>2.2.0</version>

<exclusions>

<exclusion>

<groupId>com.jayway.restassured</groupId>

<artifactId>xml-path</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.commons</groupId>

<artifactId>commons-lang3</artifactId>

</exclusion>

</exclusions>

</dependency>

<!-- Spring 4 dependencies -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-binding</artifactId>

<version>1.0.6</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-web</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-core</artifactId>

<version>${spring.security.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>aopalliance</groupId>

<artifactId>aopalliance</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>commons-collections</groupId>

<artifactId>commons-collections</artifactId>

<version>3.2</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-collections4</artifactId>

<version>4.1</version>

</dependency>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-security</artifactId>

<version>${spring.ws.version}</version>

</dependency>

<!-- Web Service related Deps -->

<dependency>

<groupId>xml-security</groupId>

<artifactId>xmlsec</artifactId>

<version>1.3.0</version>

</dependency>

<dependency>

<groupId>org.apache.axis</groupId>

<artifactId>axis</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>javax.xml</groupId>

<artifactId>jaxrpc-api</artifactId>

<version>1.1</version>

</dependency>

<dependency>

<groupId>javax.ws.rs</groupId>

<artifactId>javax.ws.rs-api</artifactId>

<version>2.0.1</version>

</dependency>

<dependency>

<groupId>commons-beanutils</groupId>

<artifactId>commons-beanutils</artifactId>

<version>1.9.2</version>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>

<dependency>

<groupId>commons-configuration</groupId>

<artifactId>commons-configuration</artifactId>

<version>1.10</version>

</dependency>

<!-- Apache cxf dependencies -->

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.10</version>

</dependency>

<dependency>

<groupId>org.codehaus.woodstox</groupId>

<artifactId>stax2-api</artifactId>

<version>3.1.1</version>

<exclusions>

<exclusion>

<artifactId>stax-api</artifactId>

<groupId>javax.xml.stream</groupId>

</exclusion>

</exclusions>

</dependency>

<!-- servlet & jsp -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.0.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

</dependency>

<!-- WebserviceTemplate dependencies -->

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-core</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-xml</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-oxm</artifactId>

<version>4.1.6.RELEASE</version>

</dependency>

<dependency>

<groupId>commons-httpclient</groupId>

<artifactId>commons-httpclient</artifactId>

<version>3.1</version>

</dependency>

<!-- Common libs -->

<dependency>

<groupId>commons-discovery</groupId>

<artifactId>commons-discovery</artifactId>

<version>0.2</version>

</dependency>

<!-- Email dependencies -->

<dependency>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

<version>1.4.5</version>

</dependency>

<dependency>

<groupId>com.sun.mail</groupId>

<artifactId>mailapi</artifactId>

<version>1.5.6</version>

</dependency>

<dependency>

<groupId>com.sun.mail</groupId>

<artifactId>smtp</artifactId>

<version>1.5.4</version>

</dependency>

<!-- URL Rewriter -->

<dependency>

<groupId>org.tuckey</groupId>

<artifactId>urlrewritefilter</artifactId>

<version>4.0.3</version>

</dependency>

<!-- Logger Dependencies -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<!-- DB drivers -->

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc6</artifactId>

<version>11.1.0.7.0</version>

</dependency>

<!-- File Upload -->

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.3</version>

</dependency>

<!--3 Excel Format Dependencies -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.15</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.15</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml-schemas</artifactId>

<version>3.15</version>

</dependency>

<dependency>

<groupId>dom4j</groupId>

<artifactId>dom4j</artifactId>

<version>1.6.1</version>

<exclusions>

<exclusion>

<artifactId>xml-apis</artifactId>

<groupId>xml-apis</groupId>

</exclusion>

</exclusions>

</dependency>

<!--2 LDAP & Encryption <dependency> <groupId>org.springframework.ldap</groupId>

<artifactId>spring-ldap-core</artifactId> <version>2.0.4.RELEASE</version>

</dependency> -->

<dependency>

<groupId>org.jasypt</groupId>

<artifactId>jasypt-spring3</artifactId>

<version>1.9.2</version>

</dependency>

<dependency>

<groupId>org.jasypt</groupId>

<artifactId>jasypt</artifactId>

<version>1.9.2</version>

</dependency>

<!-- PDF Generation -->

<dependency>

<groupId>com.itextpdf</groupId>

<artifactId>itextpdf</artifactId>

<version>5.2.1</version>

</dependency>

<dependency>

<groupId>org.bouncycastle</groupId>

<artifactId>bctsp-jdk16</artifactId>

<version>1.46</version>

</dependency>

<dependency>

<groupId>org.bouncycastle</groupId>

<artifactId>bcmail-jdk16</artifactId>

<version>1.46</version>

</dependency>

<dependency>

<groupId>org.bouncycastle</groupId>

<artifactId>bcprov-jdk16</artifactId>

<version>1.46</version>

</dependency>

<!-- Apache Web Service -->

<dependency>

<groupId>org.codehaus.woodstox</groupId>

<artifactId>woodstox-core-asl</artifactId>

<version>4.4.1</version>

<exclusions>

<exclusion>

<artifactId>stax-api</artifactId>

<groupId>javax.xml.stream</groupId>

</exclusion>

</exclusions>

</dependency>

<!-- JSON libs -->

<dependency>

<groupId>net.sf.json-lib</groupId>

<artifactId>json-lib</artifactId>

<version>2.4</version>

<classifier>jdk15</classifier>

</dependency>

<dependency>

<groupId>net.sf.ezmorph</groupId>

<artifactId>ezmorph</artifactId>

<version>1.0.6</version>

</dependency>

<dependency>

<groupId>org.json</groupId>

<artifactId>json</artifactId>

<version>20151123</version>

</dependency>

<!-- JCS Dependencies -->

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons-jcs-core</artifactId>

<version>2.0-beta-1</version>

</dependency>

<dependency>

<groupId>org.apache.jcs</groupId>

<artifactId>jcs</artifactId>

<version>1.3</version>

</dependency>

<dependency>

<groupId>concurrent</groupId>

<artifactId>concurrent</artifactId>

<version>1.3.4</version>

</dependency>

<!-- Jasper Reports Dependencies -->

<dependency>

<groupId>net.sf.jasperreports</groupId>

<artifactId>jasperreports</artifactId>

<version>6.1.1</version>

<exclusions>

<exclusion>

<groupId>org.eclipse.jdt.core.compiler</groupId>

<artifactId>ecj</artifactId>

</exclusion>

<exclusion>

<groupId>stax</groupId>

<artifactId>stax</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.lucene</groupId>

<artifactId>lucene-analyzers-common</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.lucene</groupId>

<artifactId>lucene-sandbox</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.lucene</groupId>

<artifactId>lucene-queries</artifactId>

</exclusion>

<exclusion>

<groupId>javax.xml.stream</groupId>

<artifactId>stax-api</artifactId>

</exclusion>

<exclusion>

<groupId>org.olap4j</groupId>

<artifactId>olap4j</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.lucene</groupId>

<artifactId>lucene-core</artifactId>

</exclusion>

<exclusion>

<groupId>org.apache.lucene</groupId>

<artifactId>lucene-queryparser</artifactId>

</exclusion>

<exclusion>

<groupId>xml-apis</groupId>

<artifactId>xml-apis</artifactId>

</exclusion>

<exclusion>

<groupId>com.google.zxing</groupId>

<artifactId>core</artifactId>

</exclusion>

</exclusions>

</dependency>

<!-- Jackson dependencies starts -->

<dependency>

<groupId>com.lowagie</groupId>

<artifactId>itext</artifactId>

<version>2.1.7</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-core</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-annotations</artifactId>

<version>2.7.3</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-core-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>

</dependency>

<dependency>

<groupId>org.apache.ws.xmlschema</groupId>

<artifactId>xmlschema-core</artifactId>

<version>2.0.1</version>

<exclusions>

<exclusion>

<artifactId>xml-apis</artifactId>

<groupId>xml-apis</groupId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.apache.xmlbeans</groupId>

<artifactId>xmlbeans</artifactId>

<version>2.6.0</version>

<exclusions>

<exclusion>

<artifactId>stax-api</artifactId>

<groupId>stax</groupId>

</exclusion>

</exclusions>

</dependency>

<!-- jax-ws runtime from sun -->

<dependency>

<groupId>com.sun.xml.ws</groupId>

<artifactId>jaxws-rt</artifactId>

<version>2.2.8</version>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.glassfish.metro</groupId>

<artifactId>webservices-rt</artifactId>

<version>2.3</version>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.ow2.jonas.osgi</groupId>

<artifactId>saaj-ri</artifactId>

<version>5.2.0-M5</version>

</dependency>

<dependency>

<groupId>com.sun.xml.messaging.saaj</groupId>

<artifactId>saaj-impl</artifactId>

<version>1.3.16</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<finalName>RHBI</finalName>

<plugins>

<!-- maven compiler plugin definition -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>${jdk.version}</source>

<target>${jdk.version}</target>

</configuration>

</plugin>

<!-- plugin 1-maven war plugin -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>2.5</version>

</plugin>

<!-- plugin 3-maven clean goal -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-clean-plugin</artifactId>

<version>2.6.1</version>

</plugin>

<plugin>

<groupId>org.eclipse.m2e</groupId>

<artifactId>lifecycle-mapping</artifactId>

<version>1.0.0</version>

<configuration>

<lifecycleMappingMetadata>

<pluginExecutions>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<versionRange>[2.0,)</versionRange>

<goals>

<goal>resources</goal>

<goal>testResources</goal>

</goals>

</pluginExecutionFilter>

<action>

<ignore />

</action>

</pluginExecution>

</pluginExecutions>

</lifecycleMappingMetadata>

</configuration>

</plugin>

<!-- plugin 4-maven wsimport goal -->

<!-- <plugin> <groupId>org.jvnet.jax-ws-commons</groupId> <artifactId>jaxws-maven-plugin</artifactId>

<version>2.3</version> <executions> <execution> <id>basic</id> <phase>generate-sources</phase>

<goals> <goal>wsimport</goal> </goals> </execution> </executions> <configuration>

<keep>true</keep> <sourceDestDir>${basedir}\generated\java\source</sourceDestDir>

<verbose>true</verbose> <target>2.1</target> </configuration> </plugin> -->

</plugins>

</build>

</project>

***DISPATCHERSERVLET.XML***

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"* xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:task=*"http://www.springframework.org/schema/task"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/task*

*http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<mvc:annotation-driven />

<context:annotation-config />

<context:component-scan base-package=*"com.kgisl.nsure"* />

<task:annotation-driven />

<bean id=*"internalResourceViewResolver"*

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*

p:prefix=*"/WEB-INF/jsp/"* p:suffix=*".jsp"* />

<bean

class=*"org.springframework.web.servlet.handler.SimpleMappingExceptionResolver"*>

<property name=*"exceptionMappings"*>

<props>

<prop key=*"com.kgisl.nsure.web.logging.NsureException"*>redirect:/error</prop>

</props>

</property>

</bean>

<mvc:resources location=*"/resources/"* mapping=*"/resources/\*\*"*></mvc:resources>

<mvc:resources mapping=*"/layout/\*\*"* location=*"/appresources/layout/"* />

<mvc:resources mapping=*"/login/\*\*"* location=*"/appresources/login/"* />

<mvc:resources mapping=*"/app/\*\*\*"* location=*"/appresources/js/app/"* />

<mvc:resources mapping=*"/angularmain/\*\*"* location=*"/appresources/js/angularmain/"* />

<mvc:resources mapping=*"/angularservice/\*\*\*"* location=*"/appresources/js/angularservice/"* />

<mvc:resources mapping=*"/angularmod/\*\*\*"* location=*"/appresources/js/angularmodule/"* />

<bean id=*"messageSource"*

class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<property name=*"basename"* value=*"classpath:applicationresources"* />

<property name=*"defaultEncoding"* value=*"UTF-8"*></property>

</bean>

<bean id=*"multipartResolver"*

class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*>

<property name=*"maxUploadSize"* value=*"20971520"* /> <!-- 20MB -->

</bean>

<bean id=*"localeResolver"*

class=*"org.springframework.web.servlet.i18n.SessionLocaleResolver"*>

<property name=*"defaultLocale"* value=*"en"* />

</bean>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"* />

<property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:XE"* />

<property name=*"username"* value=*"system"* />

<property name=*"password"* value=*"system123"* />

</bean>

<bean id=*"txManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<tx:annotation-driven transaction-manager=*"txManager"* />

<bean id=*"standardPBEStringEncryptor"* class=*"org.jasypt.encryption.pbe.StandardPBEStringEncryptor"*>

<property name=*"config"*>

<bean class=*"org.jasypt.encryption.pbe.config.EnvironmentStringPBEConfig"*>

<property name=*"algorithm"* value=*"PBEWithMD5AndDES"* />

<property name=*"passwordEnvName"* value=*"APP\_ENCRYPTION\_PASSWORD"* />

</bean>

</property>

</bean>

<!-- <bean class="com.kgisl.nsure.commonutils.domain.LDAPAuthenticator">

<constructor-arg ref="standardPBEStringEncryptor" /> <constructor-arg> <value>org.jasypt.encryption.pbe.StandardPBEStringEncryptor</value>

</constructor-arg> </bean> -->

<!-- <bean class="org.springframework.web.servlet.view.XmlViewResolver">

<property name="location"> <value>/WEB-INF/excel-config.xml</value> </property>

<property name="order" value="0" /> </bean> -->

<bean

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"systemPropertiesModeName"* value=*"SYSTEM\_PROPERTIES\_MODE\_OVERRIDE"* />

<property name=*"properties"*>

<bean class=*"org.apache.commons.configuration.ConfigurationConverter"*

factory-method=*"getProperties"*>

<constructor-arg>

<bean class=*"org.apache.commons.configuration.DatabaseConfiguration"*>

<constructor-arg>

<ref bean=*"dataSource"* />

</constructor-arg>

<constructor-arg value=*"EIS\_MST\_DOMAIN\_CONFIG"* /> <!-- DB Table -->

<constructor-arg value=*"VCH\_CONFIG\_NAME"* /> <!-- DB Key Column -->

<constructor-arg value=*"VCH\_CONFIG\_VALUE"* /> <!-- DB Value Column -->

</bean>

</constructor-arg>

</bean>

</property>

</bean>

<!-- <bean id="PG\_URL" class="java.lang.String"> <constructor-arg value="${migs.url}~~${migs.access.code}~~${migs.merchant.code}~~${migs.secure.hash}"

/> </bean> <bean id="MAIL\_FROM\_ADDRESS" class="com.kgisl.nsure.commonutils.domain.GlobalNames">

<property name="mailFrom" value="${mail.sender}" /> </bean> -->

<bean id=*"masterRegionCache"* class=*"org.apache.jcs.JCS"*

factory-method=*"getInstance"*>

<constructor-arg>

<value>masterRegionCache</value>

</constructor-arg>

</bean>

<bean id=*"transactionRegionCache"* class=*"org.apache.jcs.JCS"*

factory-method=*"getInstance"*>

<constructor-arg>

<value>transactionRegionCache</value>

</constructor-arg>

</bean>

<!-- Mail Template -->

<!-- <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">

<property name="host" value="${mail.smtp.host}" /> <property name="port"

value="${mail.smtp.port}" /> <property name="username" value="${mail.sender}"

/> -->

<!-- The name of the property, following JavaBean naming conventions -->

<!-- <property name="javaMailProperties"> <props> <prop key="mail.smtp.auth">${mail.smtp.auth}</prop>

<prop key="mail.debug">${mail.debug}</prop> </props> </property> </bean> -->

<!-- <bean id="wsAuthValidator" class="com.kgisl.nsure.diamond.services.security.WsAuthenticationValidator">

<property name="wsAuthService" ref="wsAuthServiceImpl"/> </bean> -->

<!-- Contains Spring WS related configurations -->

<!-- <import resource="spring-ws.xml" /> <bean id="externalAPIServiceImpl"

class="com.kgisl.nsure.externalapi.services.ExternalAPIServiceImpl" /> <bean

id="ancillaryServiceImpl" class="com.kgisl.nsure.ancillary.services.AncillaryServiceImpl"

/> <bean id="motorDetariffServiceImpl" class="com.kgisl.nsure.detariff.motor.services.MotorDetariffServiceImpl"

/> <bean id="motorServiceImpl" class="com.kgisl.nsure.transaction.services.MotorServiceImpl"

/> <bean id="administrationServiceImpl" class="com.kgisl.nsure.admin.services.AdministrationServiceImpl"

/> <bean id="clientProfileServiceImpl" class="com.kgisl.nsure.transaction.services.ClientProfileServiceImpl"

/> <bean id="premiumServiceImpl" class="com.kgisl.nsure.diamond.services.PremiumServiceImpl">

<property name="externalAPIService" ref="externalAPIServiceImpl"/> <property

name="motorDetariffService" ref="motorDetariffServiceImpl"/> <property name="motorService"

ref="motorServiceImpl"/> <property name="administrationService" ref="administrationServiceImpl"/>

<property name="clientProfileService" ref="clientProfileServiceImpl"/> <property

name="webServiceTemplate" ref="webServiceTemplate"/> </bean> <bean id="mediatorServiceImpl"

class="com.kgisl.nsure.mediator.services.MediatorServiceImpl" /> <bean id="portfolioServiceImpl"

class="com.kgisl.nsure.diamond.services.PortfolioServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="motorProposalServiceImpl"

class="com.kgisl.nsure.diamond.services.MotorProposalServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="mpaProposalServiceImpl"

class="com.kgisl.nsure.diamond.services.MpaProposalServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="motorNewProposalServiceImpl"

class="com.kgisl.nsure.diamond.services.MotorNewProposalServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="mpaPremiumCalcServiceImpl"

class="com.kgisl.nsure.diamond.services.MpaPremiumCalcServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="mpaNewProposalServiceImpl"

class="com.kgisl.nsure.diamond.services.MpaNewProposalServiceImpl"> <property

name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean id="motorRenewalProposalServiceImpl"

class="com.kgisl.nsure.diamond.services.MotorRenewalProposalServiceImpl">

<property name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean

id="hhhoProposalServiceImpl" class="com.kgisl.nsure.diamond.services.HHHOProposalServiceImpl">

<property name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean

id="hhhoNewProposalServiceImpl" class="com.kgisl.nsure.diamond.services.HHHONewProposalServiceImpl">

<property name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean

id="nmRenewalProposalServiceImpl" class="com.kgisl.nsure.diamond.services.NMRenewalProposalServiceImpl">

<property name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean

id="hhhoPremiumCalcServiceImpl" class="com.kgisl.nsure.diamond.services.HHHOPremiumCalcServiceImpl">

<property name="mediatorService" ref="mediatorServiceImpl"/> </bean> <bean

id="nsureLogger" class="com.kgisl.nsure.web.logging.NsureLogger"> <property

name="externalAPIService" ref="externalAPIServiceImpl" /> <property name="ancillaryService"

ref="ancillaryServiceImpl" /> </bean> -->

</beans>

***WEB.XML***

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"*

id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>Nsure</display-name>

<distributable></distributable>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

<async-supported>true</async-supported>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>GeneratePdf</servlet-name>

<servlet-class>com.kgisl.nsure.corresletters.utilities.GeneratePdf</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>GeneratePdf</servlet-name>

<url-pattern>/letters/printpdf</url-pattern>

</servlet-mapping>

<!-- <servlet-mapping> <servlet-name>default</servlet-name> <url-pattern>/img/\*</url-pattern>

</servlet-mapping> <servlet-mapping> <servlet-name>default</servlet-name>

<url-pattern>/images/\*</url-pattern> </servlet-mapping> -->

<servlet>

<servlet-name>FilterGeneratePdf</servlet-name>

<servlet-class>com.kgisl.nsure.filter.utilities.FilterGeneratePdf</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>FilterGeneratePdf</servlet-name>

<url-pattern>/filter/printpdf</url-pattern>

</servlet-mapping>

<!-- GZip Ends -->

<!-- Cache HTTP Starts <filter> <filter-name>jsCache</filter-name> <filter-class>com.kgisl.nsure.web.interceptor.CacheFilter</filter-class>

<init-param> <param-name>private</param-name> <param-value>false</param-value>

</init-param> <init-param> <param-name>expirationTime</param-name> <param-value>0</param-value>

</init-param> </filter> <filter-mapping> <filter-name>jsCache</filter-name>

<url-pattern>\*</url-pattern> </filter-mapping> -->

<!-- Cache HTTP Ends -->

<filter>

<filter-name>SessionFilter</filter-name>

<filter-class>com.kgisl.nsure.web.filter.SessionFilter</filter-class>

<init-param>

<param-name>avoid-urls</param-name>

<param-value>index.jsp</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>SessionFilter</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<filter>

<filter-name>UrlRewriteFilter</filter-name>

<filter-class>org.tuckey.web.filters.urlrewrite.UrlRewriteFilter</filter-class>

<init-param>

<param-name>logLevel</param-name>

<param-value>WARN</param-value>

</init-param>

<init-param>

<param-name>confReloadCheckInterval</param-name>

<param-value>1</param-value>

</init-param>

<init-param>

<param-name>confPath</param-name>

<param-value>/WEB-INF/urlrewrite.xml</param-value>

</init-param>

<init-param>

<param-name>modRewriteConf</param-name>

<param-value>false</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>UrlRewriteFilter</filter-name>

<url-pattern>/\*</url-pattern>

<dispatcher>REQUEST</dispatcher>

<dispatcher>FORWARD</dispatcher>

</filter-mapping>

<session-config>

<session-timeout>30</session-timeout>

</session-config>

<listener>

<listener-class>com.sun.xml.ws.transport.http.servlet.WSServletContextListener</listener-class>

</listener>

<servlet>

<servlet-name>WSServlet</servlet-name>

<servlet-class>com.sun.xml.ws.transport.http.servlet.WSServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>WSServlet</servlet-name>

<url-pattern>/services/\*</url-pattern>

</servlet-mapping>

</web-app>

***INDEX.JS***

**var** app = angular.module('app', [ 'ngTouch', 'ui.grid', 'ui.grid.selection' ]);

app.controller('covernoteGrid', [

'$scope',

'$http',

'$log',

'$timeout',

'uiGridConstants',

**function**($scope, $http, $log, $timeout, uiGridConstants) {

$scope.gridOptions = {

rowHeight : 25,

multiSelect : **false**,

showGridFooter : **true**,

enableSelectAll : **true**,

enableFullRowSelection : **true**,

selectionRowHeaderWidth : 35

};

$scope.gridOptions.columnDefs = [ {

name : 'NUM',

displayName : 'NO'

}, {

name : 'INTEREST\_CODE',

displayName : 'INTEREST\_CODE'

}, {

name : 'INTEREST\_DESC',

displayName : 'INTEREST\_DESCRIPTION'

}, {

name : 'SUM\_INSURED',

displayName : 'SUM\_INSURED'

}, {

name : 'RATE',

displayName : 'RATE'

}, {

name : 'PREMIUM',

displayName : 'PREMIUM'

} ];

$http.get('covernoteGrid').success(

**function**(data) {

$scope.gridOptions.data = data;

$timeout(**function**() {

**if** ($scope.gridApi.selection.selectRow) {

$scope.gridApi.selection

.selectRow($scope.gridOptions.data[0]);

}

});

});

} ]);