**Travel Premium Calculation Process**

1. **Controller Class**

package com.maan.travel.controller;

import java.util.Collections;

import java.util.List;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

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

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

import com.maan.travel.repository.VisTravelRatingRepository;

import com.maan.travel.req.PremiumCalculationReq;

import com.maan.travel.req.PrintReqService;

import com.maan.travel.res.CommonRes;

import com.maan.travel.res.PremiumCalculationRes;

import com.maan.travel.service.PremiumCalculationService;

import com.maan.travel.serviceimpl.PremiumCalculationServiceImpl;

@RestController

@RequestMapping("/api")

public class PremiumCalculationController {

@Autowired

private PremiumCalculationService entityService;

@Autowired

private PrintReqService reqPrinter;

@Autowired

private VisTravelRatingRepository repository;

@PostMapping("/premiumcalculation")

public ResponseEntity<CommonRes> getPremiumCalculation(@RequestBody PremiumCalculationReq req) {

CommonRes data = new CommonRes();

// Get All

List<PremiumCalculationRes> res = entityService.getPremiumCalculation(req);

data.setCommonResponse(res);

data.setIsError(false);

data.setErrorMessage(Collections.emptyList());

data.setMessage("Success");

if (res != null) {

return new ResponseEntity<CommonRes>(data, HttpStatus.CREATED);

} else {

return new ResponseEntity<>(null, HttpStatus.BAD\_REQUEST);

}

}

}

**2 Service Interface**

package com.maan.travel.service;

import java.util.List;

import com.maan.travel.req.PremiumCalculationReq;

import com.maan.travel.res.PremiumCalculationRes;

public interface PremiumCalculationService {

List<PremiumCalculationRes> getPremiumCalculation(PremiumCalculationReq req);

}

1. **Service Implementation Class**

package com.maan.travel.serviceimpl;

import java.math.BigDecimal;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

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

import org.springframework.stereotype.Service;

import com.google.gson.Gson;

import com.maan.travel.entity.VisTravelRatingEntity;

import com.maan.travel.repository.VisTravelRatingRepository;

import com.maan.travel.req.PremiumCalculationReq;

import com.maan.travel.res.PremiumCalculationRes;

import com.maan.travel.service.PremiumCalculationService;

@Service

public class PremiumCalculationServiceImpl implements PremiumCalculationService {

@Autowired

private VisTravelRatingRepository repository;

Gson json =new Gson();

private Logger log = LogManager.getLogger(PremiumCalculationServiceImpl.class);

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

@Override

public List<PremiumCalculationRes> getPremiumCalculation(PremiumCalculationReq req) {

// TODO Auto-generated method stub

List<PremiumCalculationRes> resList = new ArrayList<PremiumCalculationRes>();

try {

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

int ageFrom = 0;

int ageTo = 0;

int totalMembers =0;

// 1 to 4

ageFrom = 1;

ageTo = 4;

totalMembers = (Integer.*valueOf*(req.getAgeDetails().getOneToFour() ==**null**?0:Integer.*valueOf*(req.getAgeDetails().getOneToFour())));

VisTravelRatingEntity oneToFourRating = repository.findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(req.getTypeId(),**new** BigDecimal(req.getCoverId()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal(ageFrom) ,**new** BigDecimal(ageTo), req.getAgencyCode() );

BigDecimal oneToFour = oneToFourRating.getRating() ;

BigDecimal totalMembersRatingOne = **new** BigDecimal( totalMembers ).multiply(oneToFour);

// 5 to 65

ageFrom = 5;

ageTo = 65;

totalMembers = (Integer.*valueOf*(req.getAgeDetails().getFiveToSixtyFive() ==**null**?0: Integer.*valueOf*(req.getAgeDetails().getFiveToSixtyFive())));

VisTravelRatingEntity fiveToSixtyFiveRating = repository.findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(req.getTypeId(),**new** BigDecimal(req.getCoverId()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal(ageFrom) ,**new** BigDecimal(ageTo), req.getAgencyCode() );

BigDecimal fiveToSixty = fiveToSixtyFiveRating.getRating() ;

BigDecimal totalMembersRatingTwo = **new** BigDecimal( totalMembers ).multiply(fiveToSixty);

// 66 to 70

ageFrom =66;

ageTo = 70;

totalMembers = (Integer.*valueOf*(req.getAgeDetails().getSixtySixToSeventy() ==**null**?0: Integer.*valueOf*(req.getAgeDetails().getSixtySixToSeventy())));

VisTravelRatingEntity sixtySixtoSeventyRating = repository.findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(req.getTypeId(),**new** BigDecimal(req.getCoverId()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal(ageFrom) ,**new** BigDecimal(ageTo), req.getAgencyCode() );

BigDecimal sixtysixToSeventy = sixtySixtoSeventyRating.getRating();

BigDecimal totalMembersRatingThree = **new** BigDecimal( totalMembers ).multiply(sixtysixToSeventy);

// 71 to 80

ageFrom =71;

ageTo = 80;

totalMembers = (Integer.*valueOf*(req.getAgeDetails().getSeventyOneEighty() ==**null**?0:Integer.*valueOf*(req.getAgeDetails().getSeventyOneEighty())));

VisTravelRatingEntity seventytoEightyRating = repository.findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(req.getTypeId(),**new** BigDecimal(req.getCoverId()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal( req.getTravelledDays()),**new** BigDecimal(ageFrom) ,**new** BigDecimal(ageTo), req.getAgencyCode() );

BigDecimal seventyToEighty = seventytoEightyRating.getRating();

BigDecimal totalMembersRatingFour = **new** BigDecimal( totalMembers ).multiply(seventyToEighty );

// Total Rating

BigDecimal totalRating = totalMembersRatingOne.add(totalMembersRatingTwo).add(totalMembersRatingThree).add(totalMembersRatingFour);

PremiumCalculationRes res = new PremiumCalculationRes();

res.setOneToFourRating(totalMembersRatingOne ==null?"": totalMembersRatingOne.toString());

res.setFiveToSixtyFiveRating(totalMembersRatingTwo ==null?"": totalMembersRatingTwo.toString());

res.setSixtySixtoSeventyRating(totalMembersRatingThree ==null?"": totalMembersRatingThree.toString());

res.setSeventytoEightyRating(totalMembersRatingFour ==null?"": totalMembersRatingFour.toString());

res.setTotalrating(totalRating.toString());

resList.add(res);

} catch (Exception e) {

e.printStackTrace();

log.info("Exception is ---> " + e.getMessage());

return null;

}

return resList;

}

}

**4. Repository Interface**

package com.maan.travel.repository;

import java.math.BigDecimal;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.JpaSpecificationExecutor;

import com.maan.travel.entity.VisTravelRatingEntity;

import com.maan.travel.req.AgeTypeReq;

public interface VisTravelRatingRepository extends JpaRepository<VisTravelRatingEntity,BigDecimal > , JpaSpecificationExecutor<VisTravelRatingEntity>{

// List<VisTravelRatingEntity> findByTypeIdAndCoverIdAndTravelStartDateGreaterThanEqualAndTravelEndDateLessThanEqualAndAgeFromGreaterThanAndAgeToLessThanEqualAndAgencyCode(String typeId, BigDecimal coverId,Date travelStartDate, Date travelEndDate, BigDecimal ageFrom, BigDecimal ageTo, String agencyCode );

VisTravelRatingEntity findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(

String typeId, BigDecimal coverId, BigDecimal travelStartDate, BigDecimal travelEndDate,

BigDecimal ageFrom, BigDecimal ageTo, String agencyCode);

/\*

List<VisTravelRatingEntity> findByTypeIdAndCoverIdAndTravelStartDateLessThanEqualAndTravelEndDateGreaterThanEqualAndAgeFromLessThanEqualAndAgeToGreaterThanEqualAndAgencyCode(

String typeId, BigDecimal coverId, BigDecimal travelDays, String agencyCode,

AgeTypeReq ageDetails); \*/

}

**5 Outer Request Class**

package com.maan.travel.req;

import java.util.List;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.AllArgsConstructor;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Getter

@Setter

@NoArgsConstructor

@AllArgsConstructor

public class PremiumCalculationReq {

@JsonProperty("TypeId")

private String typeId ;

@JsonProperty("CoverId")

private String coverId ;

@JsonProperty("TravelledDays")

private String travelledDays ;

@JsonProperty("AgencyCode")

private String agencyCode ;

@JsonProperty("AgeDetails")

private AgeTypeReq ageDetails;

}

**6 Inner Request Class**

package com.maan.travel.req;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.AllArgsConstructor;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Getter

@Setter

@NoArgsConstructor

@AllArgsConstructor

public class AgeTypeReq {

@JsonProperty("OneToFour")

private String oneToFour;

@JsonProperty("FiveToSixtyFive")

private String fiveToSixtyFive ;

@JsonProperty("SixtySixToSeventy")

private String sixtySixToSeventy ;

@JsonProperty("SeventyOneToEighty")

private String SeventyOneEighty ;

}

**7 Response Class**

package com.maan.travel.res;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.AllArgsConstructor;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

@Getter

@Setter

@NoArgsConstructor

@AllArgsConstructor

public class PremiumCalculationRes {

@JsonProperty("OneToFourRating")

private String oneToFourRating ;

@JsonProperty("FiveToSixtyFiveRating")

private String fiveToSixtyFiveRating ;

@JsonProperty("SixtySixtoSeventyRating")

private String sixtySixtoSeventyRating ;

@JsonProperty("SeventytoEightyRating")

private String seventytoEightyRating ;

@JsonProperty("TotalRating")

private String totalrating ;

}

**8 Entity Class**

package com.maan.travel.entity;

import java.math.BigDecimal;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import org.hibernate.annotations.DynamicInsert;

import org.hibernate.annotations.DynamicUpdate;

import org.springframework.data.jpa.domain.AbstractPersistable;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

import lombok.ToString;

@Getter

@Setter

@AllArgsConstructor

@ToString

@Entity

@DynamicInsert

@DynamicUpdate

@Builder

@NoArgsConstructor

@Table(name="VIS\_TRAVEL\_RATING")

public class VisTravelRatingEntity {

@Id

@Column(name="SNO")

private BigDecimal sno ;

@Column(name="TYPE\_ID", length=5)

private String typeId ;

@Column(name="TYPE\_DESC", length=50)

private String typeDesc ;

@Column(name="COVER\_ID")

private BigDecimal coverId ;

@Column(name="COVER\_DESC", length=50)

private String coverDesc ;

@Column(name="SCHEME\_ID")

private BigDecimal schemeId ;

@Column(name="SCHEME\_DESC", length=30)

private String schemeDesc ;

@Column(name="TRAVEL\_START\_DATE")

private BigDecimal travelStartDate ;

@Column(name="TRAVEL\_END\_DATE")

private BigDecimal travelEndDate ;

@Column(name="AGE\_FROM")

private BigDecimal ageFrom ;

@Column(name="AGE\_TO")

private BigDecimal ageTo ;

@Column(name="RATING")

private BigDecimal rating ;

@Column(name="VAT")

private BigDecimal vat ;

@Column(name="TOTAL\_RATE")

private BigDecimal totalRate ;

@Column(name="AGENCY\_CODE", length=25)

private String agencyCode ;

@Column(name="VAT2")

private BigDecimal vat2 ;

@Column(name="TOTAL\_VAT")

private BigDecimal totalVat ;

@Column(name="AMEND\_ID")

private BigDecimal amendId ;

@Temporal(TemporalType.DATE)

@Column(name="EFFECTIVE\_DATE", nullable=false)

private Date effectiveDate ;

@Column(name="REMARKS", length=300)

private String remarks ;

@Column(name="STATUS", length=2)

private String status ;

@Temporal(TemporalType.DATE)

@Column(name="ENTRY\_DATE", nullable=false)

private Date entryDate ;

@Column(name="TRIP\_ID")

private BigDecimal tripId ;

@Column(name="AGEBAND\_ID")

private BigDecimal agebandId ;

}

**Output**

192.168.1.15:8084/api/premiumcalculation

**Req 🡪**

{

"TypeId":"1",

"CoverId":"2",

"TravelledDays":"6",

"AgencyCode":"88888",

"AgeDetails" :    {

        "OneToFour":1,

        "FiveToSixtyFive":2,

        "SixtySixToSeventy":3,

        "SeventyOneToEighty":4

    }

}

Res 🡪

{

    "Message": "Success",

    "IsError": **false**,

    "ErrorMessage": [],

    "Result": [

        {

            "OneToFourRating": "4.42",

            "FiveToSixtyFiveRating": "8.84",

            "SixtySixtoSeventyRating": "26.58",

            "SeventytoEightyRating": "53.12",

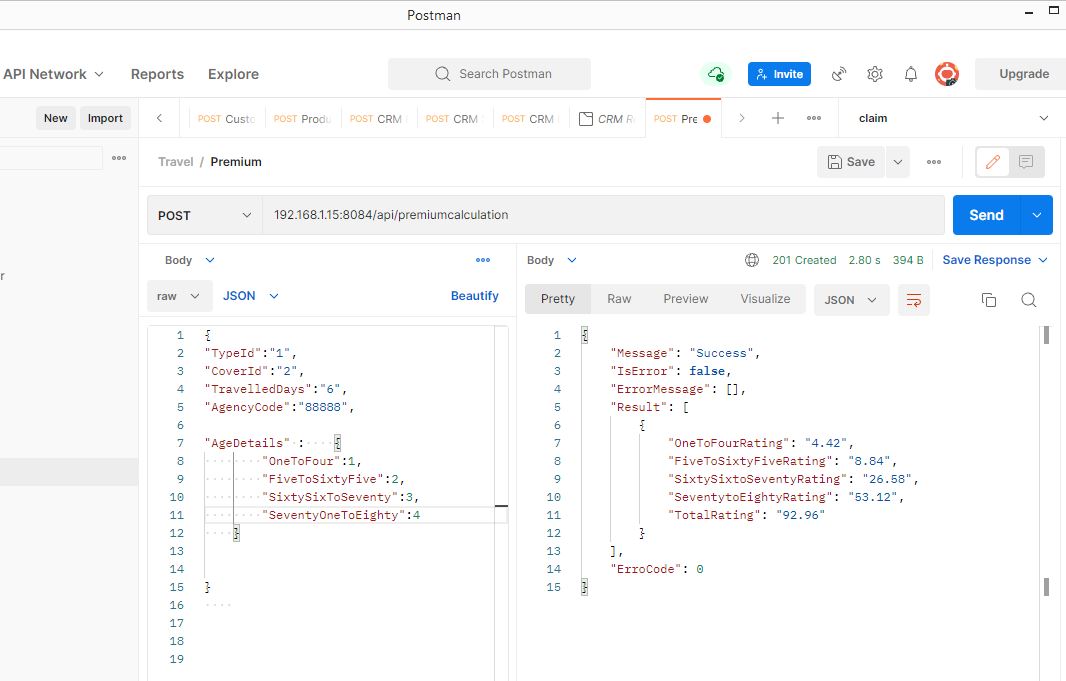
            "TotalRating": "92.96"

        }

    ],

    "ErroCode": 0

}

****