<https://www.baeldung.com/java-stream-sum> 🡪Multiple way sum

<https://www.baeldung.com/java-groupingby-collector> 🡪grouping by logic

<https://www.concretepage.com/java/jdk-8/java-8-sum-array-map-and-list-collection-example-using-reduce-and-collect-method> 🡪 function to sum

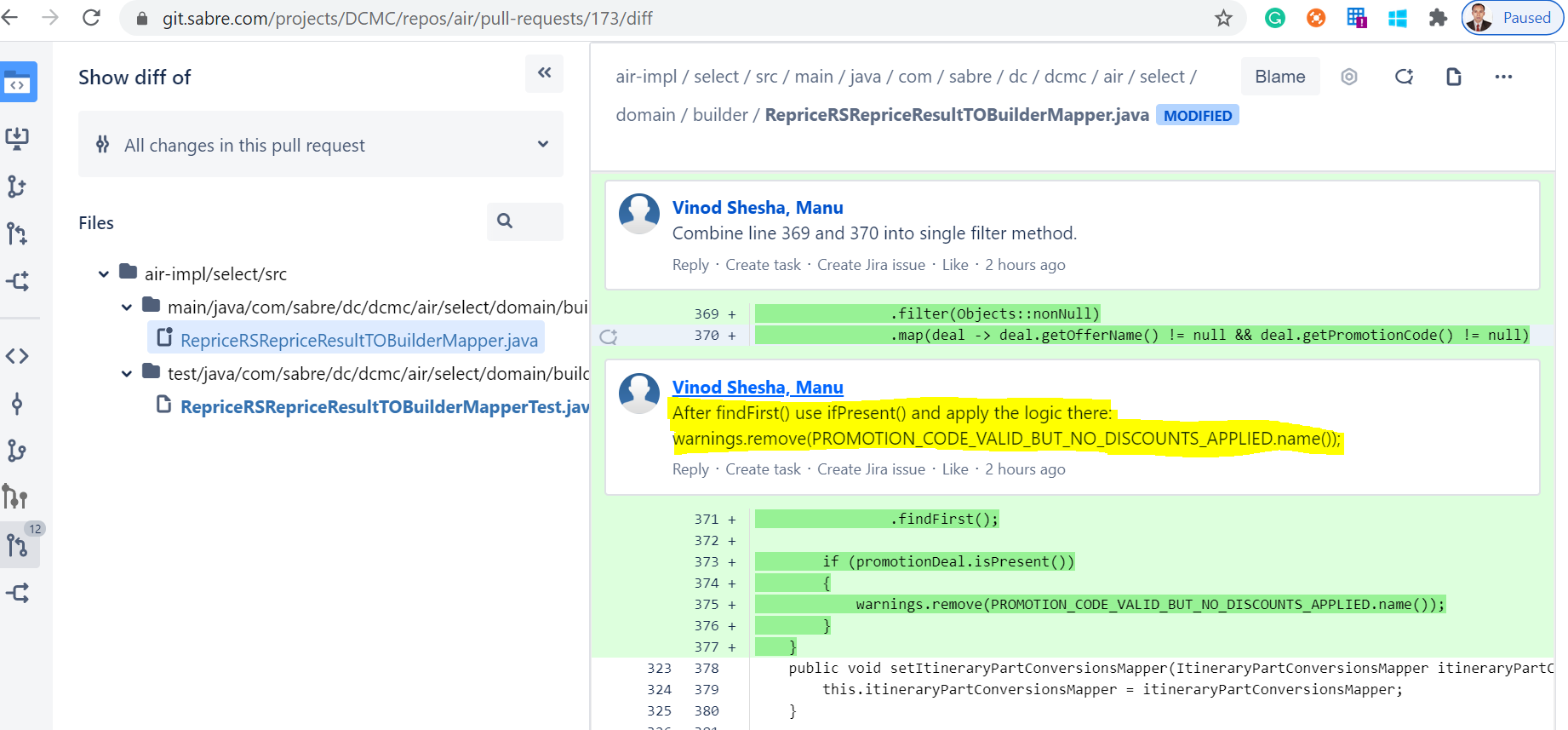
<https://zetcode.com/java/streamreduce/> 🡪Array to sum

<https://mkyong.com/java8/java-8-filter-a-map-examples/> 🡪Map Under standing



**Solution:**

|  |
| --- |
| **private void** clearNoDiscountWarningForAccountCodeViaPromoCode(Set<String> warnings, RepriceRS source) {  **long** negotiatedAccountCodeCount = 0L;  negotiatedAccountCodeCount = source.getAirPricing().getPricedItinerary()  .stream()  .map(pricedItenarary -> pricedItenarary.getItineraryPricing())  .flatMap(Collection::stream)  .map(itenararyPrcng -> itenararyPrcng.getFareBreakdown().getPassengerFareDetail())  .flatMap(Collection::stream)  .filter(fareDetail -> fareDetail.getFareInfo() != **null**)  .map(fareDetail -> fareDetail.getFareInfo().getItineraryPartFareInfo())  .flatMap(Collection::stream)  .map(ItineraryPartFareInfo::getFlightFareInfo)  .flatMap(Collection::stream)  .map(FlightFareInfo::getNegotiatedFareSource)  .filter(fareSource -> Objects.*nonNull*(fareSource) && fareSource.getAccountCode() != **null** && fareSource.getAccountCode().trim().length() > 0)  .count();  **if** (negotiatedAccountCodeCount > 0L)  {  warnings.remove(***PROMOTION\_CODE\_VALID\_BUT\_NO\_DISCOUNTS\_APPLIED***.name());  } } |

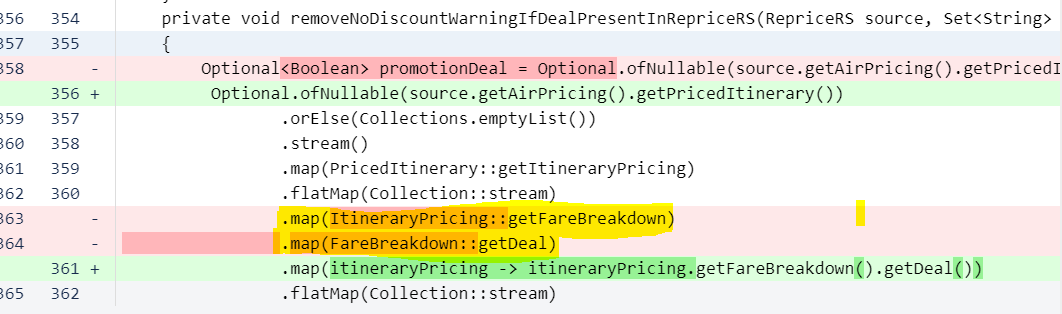


**Solution:**

|  |
| --- |
| **private void** removeNoDiscountWarningIfDealPresentInRepriceRS(RepriceRS source, Set<String> warnings) {  Optional.*ofNullable*(source.getAirPricing().getPricedItinerary())  .orElse(Collections.*emptyList*())  .stream()  .map(PricedItinerary::getItineraryPricing)  .flatMap(Collection::stream)  .map(ItineraryPricing::getFareBreakdown)  .map(FareBreakdown::getDeal)  .flatMap(Collection::stream)  .map(deal -> Objects.*nonNull*(deal) && deal.getOfferName() != **null** && deal.getPromotionCode() != **null**)  .findFirst().ifPresent(e->warnings.remove(***PROMOTION\_CODE\_VALID\_BUT\_NO\_DISCOUNTS\_APPLIED***.name())); } |

|  |
| --- |
| **private** List<Setting> loadSettings(DbSession dbSession, Optional<ComponentDto> component, Set<String> keys) {  *// List of settings must be kept in the following orders : default -> global -> component -> branch*  List<Setting> settings = **new** ArrayList<>();  settings.addAll(loadDefaultValues(keys));  settings.addAll(loadGlobalSettings(dbSession, keys));  **if** (component.[isPresent](https://www.codota.com/code/java/methods/java.util.Optional/isPresent)() && component.[get](https://www.codota.com/code/java/methods/java.util.Optional/get)().getBranch() != null && component.[get](https://www.codota.com/code/java/methods/java.util.Optional/get)().getMainBranchProjectUuid() != null) {  ComponentDto project = dbClient.componentDao().selectOrFailByUuid(dbSession, component.[get](https://www.codota.com/code/java/methods/java.util.Optional/get)().getMainBranchProjectUuid());  settings.addAll(loadComponentSettings(dbSession, keys, project).values());  }  component.[**ifPresent**](https://www.codota.com/code/java/methods/java.util.Optional/ifPresent)(componentDto -> settings.addAll(loadComponentSettings(dbSession, keys, componentDto).values()));  **return** settings.[stream](https://www.codota.com/code/java/methods/java.util.List/stream)()  .[filter](https://www.codota.com/code/java/methods/java.util.stream.Stream/filter)(s -> settingsWsSupport.isVisible(s.getKey(), s.getDefinition(), component))  .collect(Collectors.toList());  } |

**How to do multiple map to single map**



**Solution:**

|  |
| --- |
| **private void** removeNoDiscountWarningIfDealPresentInRepriceRS(RepriceRS source, Set<String> warnings) {  Optional.*ofNullable*(source.getAirPricing().getPricedItinerary())  .orElse(Collections.*emptyList*())  .stream()  .map(PricedItinerary::getItineraryPricing)  .flatMap(Collection::stream)  .map(itineraryPricing -> itineraryPricing.getFareBreakdown().getDeal())  .flatMap(Collection::stream)  .filter(deal -> Objects.*nonNull*(deal) && deal.getOfferName() != **null** && deal.getPromotionCode() != **null**)  .findFirst().ifPresent(e -> warnings.remove(***PROMOTION\_CODE\_VALID\_BUT\_NO\_DISCOUNTS\_APPLIED***.name())); } |

**Filter Example:**

|  |
| --- |
| [**https://dzone.com/articles/single-filter-perform-better-than-multiple-one-in**](https://dzone.com/articles/single-filter-perform-better-than-multiple-one-in)  public class MyBenchmark {  @Benchmark  @BenchmarkMode(Mode.All)  @OutputTimeUnit(TimeUnit.SECONDS)  public long testStreamWithSingleFilter() {  List<Double> doubles = new Random().doubles(1\_000, 1, 4).boxed().collect(Collectors.toList());  long count = doubles  .stream()  .filter(d -> d < Math.PI  && d > Math.E  && d != 3.10040970053377777  && d != 2.96240970053377777)  .count();    return count;  }  @Benchmark  @BenchmarkMode(Mode.All)  @OutputTimeUnit(TimeUnit.SECONDS)  public long testStreamWithMultipleFilter() {  List<Double> doubles = new Random().doubles(1\_000, 1, 4).boxed().collect(Collectors.toList());  long count = doubles  .stream()  .filter(d -> d > Math.E)  .filter(d -> d < Math.PI)  .filter(d -> d != 3.10040970053377777)  .filter(d -> d != 2.96240970053377777)  .count();    return count;  }  } |

<https://www.baeldung.com/java-difference-map-and-flatmap>

<https://howtodoinjava.com/java8/stream-map-vs-flatmap/>

<https://jsparrow.github.io/rules/flat-map-instead-of-nested-loops.html#code-changes> --good link

<https://examples.javacodegeeks.com/core-java/java-8-filter-null-values-stream-example/> --> filter example

Example:

.filter(fareSource -> Objects.nonNull(fareSource) && fareSource.getAccountCode() != null && fareSource.getAccountCode().trim().length() > 0)

**MultiMap**

Example:

|  |
| --- |
| private void supplementGroupedList(List<BookingClassMarketTypeGroupedSectors> groupedSectors, Map.Entry<Sector, List<AncillaryOffer>> entry)  {  for (AncillaryOffer offer : entry.getValue())  {  String bookingClass = bookingClassTranslator.translate(offer.getSubcode());  String marketType = marketTypeTranslator.translate(offer.getSubcode());  Optional<BookingClassMarketTypeGroupedSectors> bookingClassMarketTypeGroupedSectors = getCorrespondingGroup(bookingClass, groupedSectors);  if (bookingClassMarketTypeGroupedSectors.isPresent())  {  bookingClassMarketTypeGroupedSectors.get().getMarketTypeToSectorsMultiMap().put(marketType, entry.getKey());  BookingClassMarketTypeGroupedSectors bookingClassMarketTypeGroupedSectorsElement = bookingClassMarketTypeGroupedSectors.get();  Multimap<String, Sector> sectorMap = bookingClassMarketTypeGroupedSectorsElement.getMarketTypeToSectorsMultiMap();  Collection<Sector> sectors = sectorMap.get(marketType);  if (sectors != null && !sectors.contains(entry.getKey()))  { bookingClassMarketTypeGroupedSectorsElement.getMarketTypeToSectorsMultiMap().put(marketType, entry.getKey());  }  }  else  {  groupedSectors.add(new BookingClassMarketTypeGroupedSectors(bookingClass, marketType, entry.getKey()));  }  }  } |

How to get the value from MAP and do the stream operation

|  |
| --- |
| ShoppingBasket.java  **public** Map<String, RefundPaymentsWithTotal> getEmdRefundPaymentsWithTotalMap() {  **return this**.emdRefundPaymentsWithTotalMap; }  RefundEMDToBTFirstChecker**.java**  **private boolean** hasEmdToRefundAndTravelBankRefundTargets(ShoppingBasket shoppingBasket) {   **return** shoppingBasket.getEmdRefundPaymentsWithTotalMap() != **null** && !shoppingBasket.getEmdRefundPaymentsWithTotalMap().isEmpty()  && shoppingBasket.getEmdRefundPaymentsWithTotalMap().values().stream()  .allMatch(refundPaymentsWithTotal -> Optional.*ofNullable*(refundPaymentsWithTotal  .getPayments())  .orElse(**new** ArrayList<>())  .stream()  .allMatch(Payment::isBTPayment)); } |

How to work AnyMatch

|  |
| --- |
| **private boolean** hasTravelBankPayment(ShoppingBasket shoppingBasket) {  **return** shoppingBasket.getPayments() != **null** && !shoppingBasket.getPayments().isEmpty() && shoppingBasket.getPayments().size() <= 2  && shoppingBasket.getPayments().stream().anyMatch(Payment::isBTPayment); } |

How to work all match without two for loop in one line code

|  |
| --- |
| **private boolean** isSameCurrency(ShoppingBasket shoppingBasket) {   List<Payment> payments = shoppingBasket.getEmdRefundPaymentsWithTotalMap().values().stream()  .flatMap(refundPaymentsWithTotal -> refundPaymentsWithTotal.getPayments().stream()).collect(Collectors.*toList*());  **return** shoppingBasket.getPayments().stream()  .allMatch(payment -> payments.stream()  .allMatch(payment1 -> Objects.*equals*(payment1.getCurrency(), payment.getCurrency()))); } |

How to Work **.ifPresent**

|  |
| --- |
| public void apply(ConfigInitRequest request, ConfigInitResult result)  {  Optional.ofNullable(request.getChanges())  .map(ConfigChanges::getPassengerTypes)  .filter(Objects::nonNull)  .map(passengerTypeConfigStructMapper::map)  .ifPresent(passengerTypeTreeLikeConfigStructure ->  {  passengerTypeConfigProvider.setupSessionConfig(passengerTypeTreeLikeConfigStructure);  result.setPassengerTypesApplied(true);  });  } |

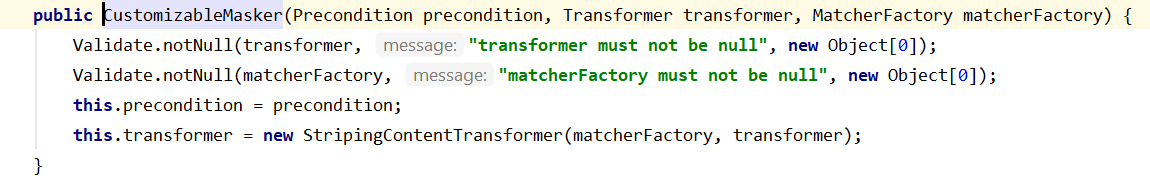
|  |
| --- |
| BreakdownElement priceBreakdown = reservationTOBreakdownElementMappers.stream()  .filter(mapper -> mapper.isApplicable(source))  .findFirst()  .get()  .map(source, new PriceComponentsMappingContext(itinerary, passengers, source.getBooking().getPassengerElectronicDocumentTOs())); |

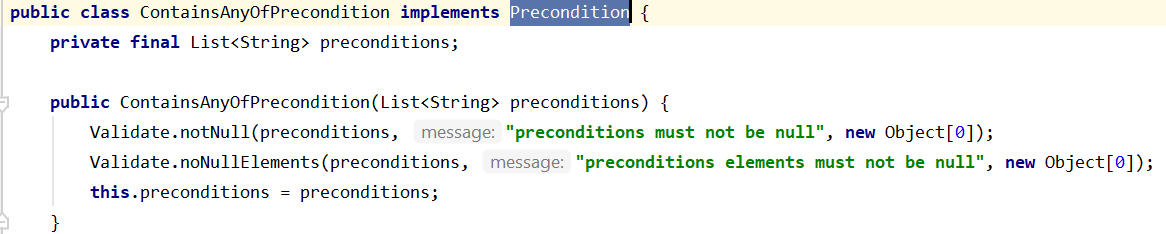
Directly .get it is bad code practice, So prefer the below code

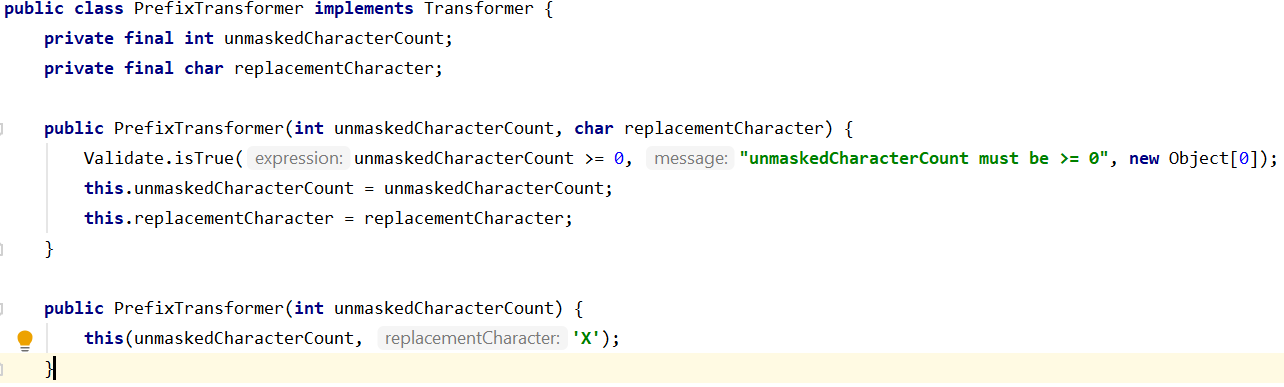
|  |
| --- |
| Optional<AbstractReservationTOBreakdownElementMapper> optional = reservationTOBreakdownElementMappers.stream()  .filter(mapper -> mapper.isApplicable(source))  .findFirst();  BreakdownElement priceBreakdown = null;  if(optional.isPresent()){  priceBreakdown= optional  .get()  .map(source, new PriceComponentsMappingContext(itinerary, passengers, source.getBooking().getPassengerElectronicDocumentTOs()));  } |

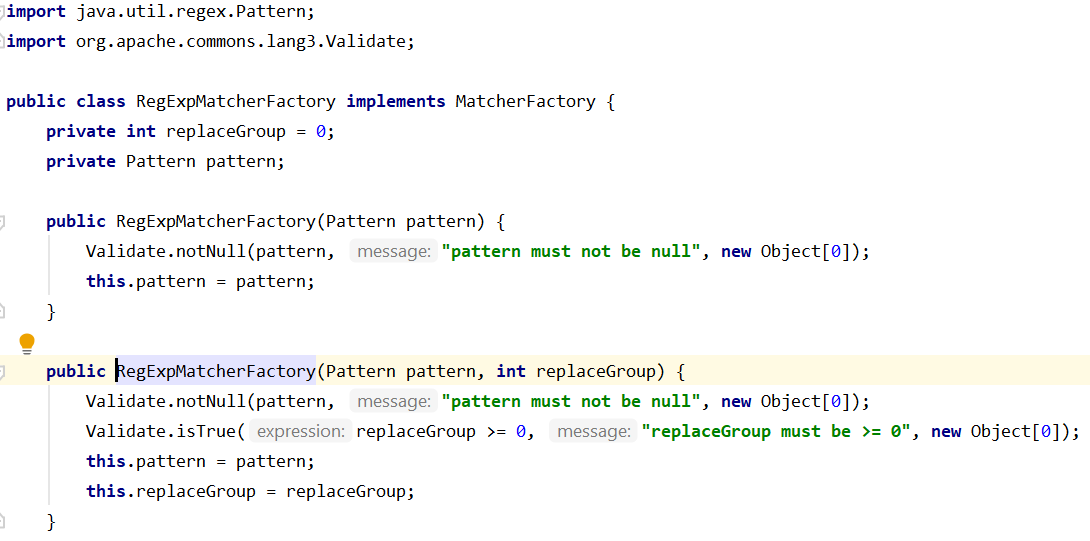
**How to create a new object of object Example:**

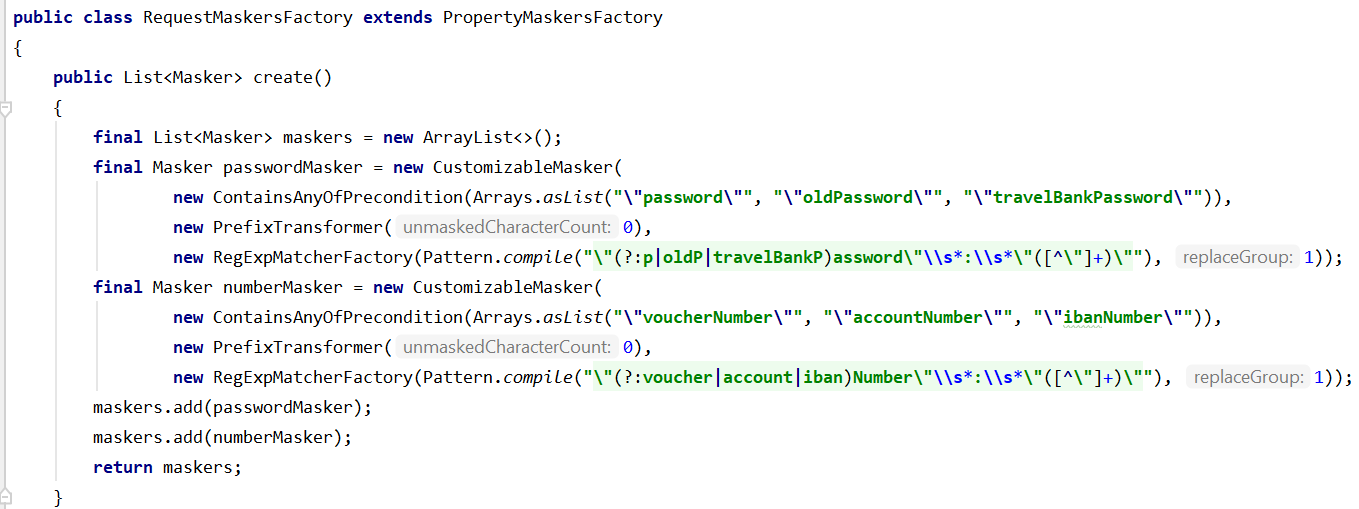












**How to use Optional.ofNullable**

|  |
| --- |
| **private** Collection<Sector> provideApplicableSectors(DiscountPriceComponent discountPriceComponent, Map<Sector, Integer> sectorToSectorNumberMap) {  ItineraryPart itineraryPartApplicableForDeal = discountPriceComponent.getPriceComponentMetadata().getItineraryPart();  **boolean** isNotPreshoppingDiscount = Optional.*ofNullable*(itineraryPartApplicableForDeal).isPresent();  **if** (isNotPreshoppingDiscount)  {  **return** itineraryPartApplicableForDeal.getSectors();  }  **return** sectorToSectorNumberMap.keySet(); } |

**How to Use method inside filter:**

**Example:**

|  |
| --- |
| **private** List<PassengerDetailType.Document> filterDocumentsPaidWithOtherFormOfPayment(List<PassengerDetailType.Document> documents, **final** Map<String, EmdDocumentAmountBreakdown> excludedEmdAmountBreakdown) {  **return** Lists.*newArrayList*(documents.stream().filter(document -> {  **boolean** shouldRetain = **true**;  **if** (excludedEmdAmountBreakdown.containsKey(document.getDocNumber()))  {  shouldRetain = !isWholeAmountCoveredInBreakdown(excludedEmdAmountBreakdown.get(document.getDocNumber()), document);  }  **return** shouldRetain;  }).collect(Collectors.*toList*())); }  **private boolean** isWholeAmountCoveredInBreakdown(EmdDocumentAmountBreakdown amount, PassengerDetailType.Document document) {  **return** amount.getBasePrice().compareTo(document.getBaseFare()) == 0 && amount.getTaxPrice().compareTo(document.getTaxes()) == 0; } |

How to MAP iterate:

|  |
| --- |
| **private int** provideFallbackSectorMatching(Map<Sector, Integer> sectorToSectorNumberMap, Sector sector) {  Set<SectorFlight.FlightEqualsOptions> simpleMatchingRulesSet = Collections.*set*(SectorFlight.FlightEqualsOptions.***DEPARTURE\_DATE\_TIME***, SectorFlight.FlightEqualsOptions.***AIRLINE***,  SectorFlight.FlightEqualsOptions.***DESTINATION***, SectorFlight.FlightEqualsOptions.***ORIGIN***);  **return** sectorToSectorNumberMap.entrySet()  .stream()  .filter(entry -> entry.getKey().getFlight().correspondsTo(sector.getFlight(), simpleMatchingRulesSet))  .map(Map.Entry::getValue)  .findFirst()  .get(); }  **public boolean** correspondsTo(IFlight sectorFlight, Set<SectorFlight.FlightEqualsOptions> options) {  EqualsBuilder sectorFlights = **this**.createEqualsBuilderFor(**this**, sectorFlight, options);  **return** sectorFlights.isEquals(); } |

How to Iterate the Map with Collection:

|  |
| --- |
| **private void** translateSectors(DiscountPriceComponent discount, Map<String, Set<String>> dealIdentityToSectorsMap, Map<Sector, Integer> sectorToSectorNumberMap) {  String dealIdentity = discount.getOfferingName();  **if** (!dealIdentityToSectorsMap.containsKey(dealIdentity))  {  dealIdentityToSectorsMap.put(dealIdentity, Sets.*newHashSet*());  }  Set<String> sectors = dealIdentityToSectorsMap.get(dealIdentity);  provideApplicableSectors(discount, sectorToSectorNumberMap).forEach(sector -> {  **int** sectorNumber = sectorToSectorNumberMap.entrySet()  .stream()  .filter(entry -> entry.getKey().equalsByFlight(sector))  .map(Map.Entry::getValue)  .findFirst()  .orElseGet(() -> provideFallbackSectorMatching(sectorToSectorNumberMap, sector));  String numberValue = String.*valueOf*(sectorNumber);  **if** (!sectors.contains(numberValue))  {  sectors.add(numberValue);  }  }); }  **private** Collection<Sector> provideApplicableSectors(DiscountPriceComponent discountPriceComponent, Map<Sector, Integer> sectorToSectorNumberMap) {  ItineraryPart itineraryPartApplicableForDeal = discountPriceComponent.getPriceComponentMetadata().getItineraryPart();  **boolean** isNotPreshoppingDiscount = Optional.*ofNullable*(itineraryPartApplicableForDeal).isPresent();  **if** (isNotPreshoppingDiscount)  {  **return** itineraryPartApplicableForDeal.getSectors();  }  **return** sectorToSectorNumberMap.keySet(); } |

How to sort a list in Jdk 1.8

|  |
| --- |
| **private void sorting(List<PaymentTO> list)**  {  java.util.Collections.sort(list, new Comparator<PaymentTO>()  {  @Override public int compare(PaymentTO paymentFirst, PaymentTO paymentSecond)  {  if (paymentFirst instanceof CreditCardPaymentTO && paymentSecond instanceof CreditCardPaymentTO)  {  CreditCardPaymentTO creditCardPaymentFirst = (CreditCardPaymentTO) paymentFirst;  int insMonthFirst = creditCardPaymentFirst.getCreditCard().getCreditCardInstallments().get(0).getInstallmentMonths();  CreditCardPaymentTO creditCardPaymentSecond = (CreditCardPaymentTO) paymentSecond;  int insMonthSecond = creditCardPaymentSecond.getCreditCard().getCreditCardInstallments().get(0).getInstallmentMonths();  return insMonthSecond - insMonthFirst;  }  return 0;  }  });  }  **Soln:**  **private void** sorting(List<PaymentTO> list) {  java.util.Collections.*sort*(list, (paymentFirst, paymentSecond) -> {  **if** (paymentFirst **instanceof** CreditCardPaymentTO && paymentSecond **instanceof** CreditCardPaymentTO)  {  CreditCardPaymentTO creditCardPaymentFirst = (CreditCardPaymentTO) paymentFirst;  **int** insMonthFirst = creditCardPaymentFirst.getCreditCard().getCreditCardInstallments().get(0).getInstallmentMonths();  CreditCardPaymentTO creditCardPaymentSecond = (CreditCardPaymentTO) paymentSecond;  **int** insMonthSecond = creditCardPaymentSecond.getCreditCard().getCreditCardInstallments().get(0).getInstallmentMonths();  **return** insMonthSecond - insMonthFirst;  }  **return** 0;  }); } |

**How to declare the empty list:**

Collections.*emptyList*();

List<String> list= Collections.*emptyList*();

**Example**:

**public** List<PassengerTO> getPassengers()  
{  
 **return booking** != **null** ? **booking**.getPassengers() : Collections.*emptyList*();  
}

**How to check null object:**

**if** (Objects.*isNull*(discount.getPercentageValue()))

**How to declare the new ArrayList:**

Lists.*newArrayList*(); or Lists.*newArrayList*(..);

**Example:**

List<PassengerTO> updatedPassengers = Lists.*newArrayList*();

**How to create new hashSet**

Sets.*newHashSet(); or* Sets.*newHashSet*(..);

*Example:*

**if** (CollectionUtils.*isNotEmpty*(promoSSRs) && promoSSRs.size() >= passengerTO.getPassengerIndex())  
{  
 preferences = preferences.withSeatsPromoSSRs(Sets.*newHashSet*((promoSSRs.get(passengerTO.getPassengerIndex() - 1).split(**","**))));  
}  
**else**{  
 preferences = preferences.withSeatsPromoSSRs(Sets.*newHashSet*());  
}

**How to create a new HashMap:**

Maps.*newHashMap*();

Map<String, String> emptyMap = Collections.emptyMap();

**How to check list is not empty check:**

CollectionUtils.*isNotEmpty*(promoSSRs)

**How to check String is empty using commons lang jar**

StringUtils.*isEmpty*(request.getDepartureDate())

**How to create a set via guava jar**

**final** Set<String> searchErrors = ImmutableSet.*of*(***INVALID\_ARGUMENT***);  
**final** Map<String, Set<String>> errors = ImmutableMap.*of*(***SELECT\_FLIGHTS***, searchErrors);  
**throw new** RequestValidationException(errors);

**How to check null object with conditional operator vs Optional**

|  |
| --- |
| public PaymentRQ createRequest(PaymentProcessingContext params)  {  ShoppingBasket shoppingBasket = params.getShoppingBasket();  paymentCCAuthRequestCreatorHelper.setCreditCardAuthorisationConfiguration(creditCardAuthorisationConfiguration);  PaymentRQ paymentRQ = paymentCCAuthRequestCreatorHelper.createPaymentRQ(PaymentServiceConstants.ACTION\_AUTH, shoppingBasket.getFraudnetData());  POSType pos = paymentCCAuthRequestCreatorHelper.createPOSType(shoppingBasket);  if (threeDSecureConfiguration.isV2Enabled())  {  CreditCardPayment creditCardPayment = (CreditCardPayment) params.getPayment();  CreditCardPaymentMethodDetail creditCardPaymentMethodDetail = creditCardPayment != null ? creditCardPayment.getCreditCardPaymentMethodDetail() : null;  CreditCard creditCard = creditCardPaymentMethodDetail != null ? creditCardPaymentMethodDetail.getCreditCard() : null;  com.sabre.ssw.model.payment.form.BrowserDetails browserDetails = creditCard != null ? creditCard.getBrowserDetails() : null;  if (browserDetails != null)  {  pos.setBrowserDetail(pos.getBrowserDetail()  .withBrowserJavaEnabled(browserDetails.getBrowserJavaEnabled())  .withBrowserJavascriptEnabled(browserDetails.getBrowserJavascriptEnabled())  .withBrowserScreenColorDepth(BigInteger.valueOf(browserDetails.getBrowserScreenColorDepth()))  .withBrowserScreenWidth(BigInteger.valueOf(browserDetails.getBrowserScreenWidth()))  .withBrowserScreenHeight(BigInteger.valueOf(browserDetails.getBrowserScreenHeight()))  .withBrowserTimeZoneOffset(BigInteger.valueOf(browserDetails.getBrowserTimeZoneOffset()))  .withChallengeWindowSize(BigInteger.valueOf(browserDetails.getChallengeWindowSize())));  }  }  paymentRQ.setPOS(pos);  MerchantDetailType merchantDetail = paymentRequestCreatorHelper.createMerchantDetail();  paymentRQ.setMerchantDetail(merchantDetail);  PaymentRQ.OrderDetail orderDetail = paymentCCAuthRequestCreatorHelper.createOrderDetail(shoppingBasket, nameRequestCreator, params.getPayment());  paymentRQ.setOrderDetail(orderDetail);  PaymentDetailType paymentDetail = paymentCCAuthRequestCreatorHelper.createPaymentDetail(shoppingBasket, (CreditCardPayment) params.getPayment());  paymentRQ.getPaymentDetail().add(paymentDetail);  return paymentRQ;  }    ==================================================================================  public PaymentRQ createRequest(PaymentProcessingContext params)  {  ShoppingBasket shoppingBasket = params.getShoppingBasket();  paymentCCAuthRequestCreatorHelper.setCreditCardAuthorisationConfiguration(creditCardAuthorisationConfiguration);  PaymentRQ paymentRQ = paymentCCAuthRequestCreatorHelper.createPaymentRQ(PaymentServiceConstants.ACTION\_AUTH, shoppingBasket.getFraudnetData());  POSType pos = paymentCCAuthRequestCreatorHelper.createPOSType(shoppingBasket);  if (threeDSecureConfiguration.isV2Enabled())  {  Optional<com.sabre.ssw.model.payment.form.BrowserDetails> browserDetailsOp = Optional.empty();  Optional<CreditCardPayment> creditCardPaymentOp = Optional.ofNullable((CreditCardPayment) params.getPayment());  if (creditCardPaymentOp.isPresent())  {  CreditCardPayment creditCardPayment = creditCardPaymentOp.get();  Optional<CreditCardPaymentMethodDetail> creditCardPaymentMethodDetailOp = Optional.ofNullable(creditCardPayment.getCreditCardPaymentMethodDetail());  if (creditCardPaymentMethodDetailOp.isPresent())  {  CreditCardPaymentMethodDetail creditCardPaymentMethodDetail = creditCardPaymentMethodDetailOp.get();  Optional<CreditCard> creditCardOp = Optional.ofNullable(creditCardPaymentMethodDetail.getCreditCard());  if (creditCardOp.isPresent())  {  CreditCard creditCard = creditCardOp.get();  browserDetailsOp = Optional.ofNullable(creditCard.getBrowserDetails());  }  }  }  if (browserDetailsOp.isPresent())  {  com.sabre.ssw.model.payment.form.BrowserDetails browserDetails = browserDetailsOp.get();  pos.setBrowserDetail(pos.getBrowserDetail()  .withBrowserJavaEnabled(browserDetails.getBrowserJavaEnabled())  .withBrowserJavascriptEnabled(browserDetails.getBrowserJavascriptEnabled())  .withBrowserScreenColorDepth(BigInteger.valueOf(browserDetails.getBrowserScreenColorDepth()))  .withBrowserScreenWidth(BigInteger.valueOf(browserDetails.getBrowserScreenWidth()))  .withBrowserScreenHeight(BigInteger.valueOf(browserDetails.getBrowserScreenHeight()))  .withBrowserTimeZoneOffset(BigInteger.valueOf(browserDetails.getBrowserTimeZoneOffset()))  .withChallengeWindowSize(BigInteger.valueOf(browserDetails.getChallengeWindowSize())));  }  }  paymentRQ.setPOS(pos);  MerchantDetailType merchantDetail = paymentRequestCreatorHelper.createMerchantDetail();  paymentRQ.setMerchantDetail(merchantDetail);  PaymentRQ.OrderDetail orderDetail = paymentCCAuthRequestCreatorHelper.createOrderDetail(shoppingBasket, nameRequestCreator, params.getPayment());  paymentRQ.setOrderDetail(orderDetail);  PaymentDetailType paymentDetail = paymentCCAuthRequestCreatorHelper.createPaymentDetail(shoppingBasket, (CreditCardPayment) params.getPayment());  paymentRQ.getPaymentDetail().add(paymentDetail);  return paymentRQ;  } |

**How to declare a empty string using StringUtils.**

String errorCode = StringUtils.EMPTY;

String resultDescription = StringUtils.EMPTY;

**Understanding Of MAP**

Before Java 8 :

Map<Integer, String> map = new HashMap<>();

map.put(1, "linode.com");

map.put(2, "heroku.com");

String result = "";

for (Map.Entry<Integer, String> entry : map.entrySet()) {

if("something".equals(entry.getValue())){

result = entry.getValue();

}

}

With Java 8, you can convert a Map.entrySet() into a stream, follow by a filter() and collect() it.

Map<Integer, String> map = new HashMap<>();

map.put(1, "linode.com");

map.put(2, "heroku.com");

*//Map -> Stream -> Filter -> String*

String result = map.entrySet().stream()

.filter(x -> "something".equals(x.getValue()))

.map(x->x.getValue())

.collect(Collectors.joining());

*//Map -> Stream -> Filter -> MAP*

Map<Integer, String> collect = map.entrySet().stream()

.filter(x -> x.getKey() == 2)

.collect(Collectors.toMap(x -> x.getKey(), x -> x.getValue()));

*// or like this*

Map<Integer, String> collect = map.entrySet().stream()

.filter(x -> x.getKey() == 3)

.collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));

1. Java 8 – Filter a Map

A full example to filter a Map by values and return a String.

TestMapFilter.java

package com.mkyong;

import java.util.HashMap;

import java.util.Map;

import java.util.stream.Collectors;

public class TestMapFilter {

public static void main(String[] args) {

Map<Integer, String> HOSTING = new HashMap<>();

HOSTING.put(1, "linode.com");

HOSTING.put(2, "heroku.com");

HOSTING.put(3, "digitalocean.com");

HOSTING.put(4, "aws.amazon.com");

*// Before Java 8*

String result = "";

for (Map.Entry<Integer, String> entry : HOSTING.entrySet()) {

if ("aws.amazon.com".equals(entry.getValue())) {

result = entry.getValue();

}

}

System.out.println("Before Java 8 : " + result);

*//Map -> Stream -> Filter -> String*

result = HOSTING.entrySet().stream()

.filter(map -> "aws.amazon.com".equals(map.getValue()))

.map(map -> map.getValue())

.collect(Collectors.joining());

System.out.println("With Java 8 : " + result);

*// filter more values*

result = HOSTING.entrySet().stream()

.filter(x -> {

if (!x.getValue().contains("amazon") && !x.getValue().contains("digital")) {

return true;

}

return false;

})

.map(map -> map.getValue())

.collect(Collectors.joining(","));

System.out.println("With Java 8 : " + result);

}

}

Output

Before Java 8 : aws.amazon.com

With Java 8 : aws.amazon.com

With Java 8 : linode.com,heroku.com

2. Java 8 – Filter a Map #2

Yet another example to filter a Map by key, but this time will return a Map

TestMapFilter2.java

package com.mkyong;

import java.util.HashMap;

import java.util.Map;

import java.util.stream.Collectors;

public class TestMapFilter2 {

public static void main(String[] args) {

Map<Integer, String> HOSTING = new HashMap<>();

HOSTING.put(1, "linode.com");

HOSTING.put(2, "heroku.com");

HOSTING.put(3, "digitalocean.com");

HOSTING.put(4, "aws.amazon.com");

*//Map -> Stream -> Filter -> Map*

Map<Integer, String> collect = HOSTING.entrySet().stream()

.filter(map -> map.getKey() == 2)

.collect(Collectors.toMap(p -> p.getKey(), p -> p.getValue()));

System.out.println(collect); *//output : {2=heroku.com}*

Map<Integer, String> collect2 = HOSTING.entrySet().stream()

.filter(map -> map.getKey() <= 3)

.collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));

System.out.println(collect2); *//output : {1=linode.com, 2=heroku.com, 3=digitalocean.com}*

}

}

Output

{2=heroku.com}

{1=linode.com, 2=heroku.com, 3=digitalocean.com}

3. Java 8 – Filter a Map #3 – Predicate

This time, try the new Java 8 Predicate

TestMapFilter3.java

package com.mkyong;

import java.util.HashMap;

import java.util.Map;

import java.util.function.Predicate;

import java.util.stream.Collectors;

public class TestMapFilter3 {

*// Generic Map filterbyvalue, with predicate*

public static <K, V> Map<K, V> filterByValue(Map<K, V> map, Predicate<V> predicate) {

return map.entrySet()

.stream()

.filter(x -> predicate.test(x.getValue()))

.collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));

}

public static void main(String[] args) {

Map<Integer, String> HOSTING = new HashMap<>();

HOSTING.put(1, "linode.com");

HOSTING.put(2, "heroku.com");

HOSTING.put(3, "digitalocean.com");

HOSTING.put(4, "aws.amazon.com");

HOSTING.put(5, "aws2.amazon.com");

*// {1=linode.com}*

Map<Integer, String> filteredMap = filterByValue(HOSTING, x -> x.contains("linode"));

System.out.println(filteredMap);

*// {1=linode.com, 4=aws.amazon.com, 5=aws2.amazon.com}*

Map<Integer, String> filteredMap2 = filterByValue(HOSTING, x -> (x.contains("aws") || x.contains("linode")));

System.out.println(filteredMap2);

*// {4=aws.amazon.com}*

Map<Integer, String> filteredMap3 = filterByValue(HOSTING, x -> (x.contains("aws") && !x.contains("aws2")));

System.out.println(filteredMap3);

*// {1=linode.com, 2=heroku.com}*

Map<Integer, String> filteredMap4 = filterByValue(HOSTING, x -> (x.length() <= 10));

System.out.println(filteredMap4);

}

}

Output

{1=linode.com}

{1=linode.com, 4=aws.amazon.com, 5=aws2.amazon.com}

{4=aws.amazon.com}

{1=linode.com, 2=heroku.com}

SeatBundleMandatoryValidator u can under stand very good in list, map and set

Null check and add all in a list in a line example

=========================================================================

List<SelectionEntry> seatSelectionEntries = Lists.newArrayList();

SeatSelectionDataProvider seatSelectionDataProvider = abstractFlow.getSeatSelectionDataProvider();

if (ProviderUtil.isPresent(seatSelectionDataProvider))

{

Optional.ofNullable(seatSelectionDataProvider.getSelectionEntries())

.ifPresent(seatSelectionEntries::addAll);

}

return seatSelectionEntries;

List<AncillaryQuantity> ancillaryQuantity=AncillariesAssignment.getAncillaryQuantities();

---------------------------good example------------------------------------------

List<PassengerUpgradePrice> passengerUpgradePrices = new ArrayList<>();

passengers.stream().forEach(passenger -> {

Map<LineItemTO, List<PriceComponentTO>> passengerSectorPriceComponentsMap = getPriceComponentsMapForPax(passengersSectorPriceComponentsMap, passenger);

if (MapUtils.isNotEmpty(passengerSectorPriceComponentsMap))

{

List<MoneyTO> passengerAncillaryComponentsPrices = getPassengerUpgradePriceComponentsPrices(passengerSectorPriceComponentsMap);

if(CollectionUtils.isNotEmpty(passengerAncillaryComponentsPrices))

{

ApiPrice passengerUpgradeAncillariesPrice = createPrice(passengerAncillaryComponentsPrices);

PassengerUpgradePrice passengerUpgradePrice = new PassengerUpgradePrice();

passengerUpgradePrice.setPassenger(passenger);

passengerUpgradePrice.setUpgradePrice(passengerUpgradeAncillariesPrice);

passengerUpgradePrices.add(passengerUpgradePrice);

}

}

});

segmentUpgradeInfo.setPassengerUpgradePrices(passengerUpgradePrices);

-----------------------------------------------------------------------------

ReservationTOPnrMapper.java

private List<PaymentTO> getPayments(ReservationTO reservation)

{

List<PaymentTO> payments = getCurrentPayments(reservation);

return payments.stream()

.filter(p -> !GIFT\_CARD\_TYPE.equals(p.getPaymentType()) || p.getAmount() != null)

.filter(p -> p.getAmount() == null || p.getAmount().isNotZero())

.collect(Collectors.toList());

}

=================================================================================================

com.kartik.ssw2010.front.web.json.api.pnr.mapping.ReservationTOPnrMapper.java

List<PnrPayment> payments = newArrayList();

for (PaymentTO paymentTO:mergedPayments) {

payments.add(paymentTOpaymentMapper.mapInternal(paymentTO));

}

if(payments.isEmpty() || payments.size()<=0) {

payments = paymentTOpaymentMapper.map(mergedPayments);

}

com.kartik.ssw2010.front.web.json.api.pnr.mapping.PaymentTOPaymentMapper.java

com.kartik.ssw2010.front.instrumentation.v12.context.common.mapper.PaymentTOPaymentMapper.java

=================================================================================================

SeatBundleMandatoryValidator u can under stand very good in list, map and set

Null check and add all in a list in a line example

=========================================================================

List<SelectionEntry> seatSelectionEntries = Lists.newArrayList();

SeatSelectionDataProvider seatSelectionDataProvider = abstractFlow.getSeatSelectionDataProvider();

if (ProviderUtil.isPresent(seatSelectionDataProvider))

{

Optional.ofNullable(seatSelectionDataProvider.getSelectionEntries())

.ifPresent(seatSelectionEntries::addAll);

}

return seatSelectionEntries;

List<AncillaryQuantity> ancillaryQuantity=AncillariesAssignment.getAncillaryQuantities();

---------------------------good example------------------------------------------

List<PassengerUpgradePrice> passengerUpgradePrices = new ArrayList<>();

passengers.stream().forEach(passenger -> {

Map<LineItemTO, List<PriceComponentTO>> passengerSectorPriceComponentsMap = getPriceComponentsMapForPax(passengersSectorPriceComponentsMap, passenger);

if (MapUtils.isNotEmpty(passengerSectorPriceComponentsMap))

{

List<MoneyTO> passengerAncillaryComponentsPrices = getPassengerUpgradePriceComponentsPrices(passengerSectorPriceComponentsMap);

if(CollectionUtils.isNotEmpty(passengerAncillaryComponentsPrices))

{

ApiPrice passengerUpgradeAncillariesPrice = createPrice(passengerAncillaryComponentsPrices);

PassengerUpgradePrice passengerUpgradePrice = new PassengerUpgradePrice();

passengerUpgradePrice.setPassenger(passenger);

passengerUpgradePrice.setUpgradePrice(passengerUpgradeAncillariesPrice);

passengerUpgradePrices.add(passengerUpgradePrice);

}

}

});

segmentUpgradeInfo.setPassengerUpgradePrices(passengerUpgradePrices);

-----------------------------------------------------------------------------

ReservationTOPnrMapper.java

private List<PaymentTO> getPayments(ReservationTO reservation)

{

List<PaymentTO> payments = getCurrentPayments(reservation);

return payments.stream()

.filter(p -> !GIFT\_CARD\_TYPE.equals(p.getPaymentType()) || p.getAmount() != null)

.filter(p -> p.getAmount() == null || p.getAmount().isNotZero())

.collect(Collectors.toList());

}

=================================================================================================

com.kartik.ssw2010.front.web.json.api.pnr.mapping.ReservationTOPnrMapper.java

List<PnrPayment> payments = newArrayList();

for (PaymentTO paymentTO:mergedPayments) {

payments.add(paymentTOpaymentMapper.mapInternal(paymentTO));

}

if(payments.isEmpty() || payments.size()<=0) {

payments = paymentTOpaymentMapper.map(mergedPayments);

}

com.kartik.ssw2010.front.web.json.api.pnr.mapping.PaymentTOPaymentMapper.java

com.kartik.ssw2010.front.instrumentation.v12.context.common.mapper.PaymentTOPaymentMapper.java

# [**Java 8 lambda for selecting top salary employee for each department**](https://stackoverflow.com/questions/36040945/java-8-lambda-for-selecting-top-salary-employee-for-each-department)

Map<String, Employee> topEmployees =

allEmployees.stream()

.collect(Collectors.groupingBy(

e -> e.department,

collectingAndThen(maxBy(comparingInt(e -> e.salary)), Optional::get)

));

<https://harshitjain.home.blog/2019/06/28/solving-real-time-queries-using-java-8-features-employee-management-system/>

**How many male and female employees are there in the organization?**

Map<String, Long> noOfMaleAndFemaleEmployees=

employeeList.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.counting()));

System.out.println(noOfMaleAndFemaleEmployees);

**Print the name of all departments in the organization?**

employeeList.stream()

            .map(Employee::getDepartment)

            .distinct()

            .forEach(System.out::println);

**Count the number of employees in each department?**

Map<String, Long> employeeCountByDepartment=

employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment, Collectors.counting()));

System.out.println(employeeCountByDepartment);

**What is the average age of male and female employees?**

Map<String, Double> avgAgeOfMaleAndFemaleEmployees=

employeeList.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.averagingInt(Employee::getAge)));

System.out.println(avgAgeOfMaleAndFemaleEmployees);

**Get the details of highest paid employee in the organization?**

Optional<Employee> highestPaidEmployeeWrapper=

employeeList.stream().collect(Collectors.maxBy(Comparator.comparingDouble(Employee::getSalary)));

Employee highestPaidEmployee = highestPaidEmployeeWrapper.get();

**What is the average salary of each department?**

Map<String, Double> avgSalaryOfDepartments=

employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment, Collectors.averagingDouble(Employee::getSalary)));

System.out.println(avgSalaryOfDepartments);

**Get the details of youngest male employee in the product development department?**

Employee youngestMaleEmployeeInProductDevelopment=

employeeList.stream()

            .filter(e -> e.getGender()=="Male" && e.getDepartment()=="Product Development")

            .min(Comparator.comparingInt(Employee::getAge)).get();

**Who has the most working experience in the organization?**

Employee seniorMostEmployee =

employeeList.stream().sorted(Comparator.comparingInt(Employee::getYearOfJoining)).findFirst().get();

// OR

Employee seniorMostEmployee = employeeList.stream().min(Comparator.comparingDouble(Employee::getYearOfJoining)).get();

**List down the names of all employees in each department?**

Map<String, List<Employee>> employeeListByDepartment=

employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment));

Set<Entry<String, List<Employee>>> entrySet = employeeListByDepartment.entrySet();

for (Entry<String, List<Employee>> entry : entrySet)

{

    System.out.println("--------------------------------------");

    System.out.println("Employees In "+entry.getKey() + " : ");

    System.out.println("--------------------------------------");

    List<Employee> list = entry.getValue();

    for (Employee e : list)

    {

        System.out.println(e.getName());

    }

}

**What is the average salary and total salary of the whole organization?**

DoubleSummaryStatistics employeeSalaryStatistics=

employeeList.stream().collect(Collectors.summarizingDouble(Employee::getSalary));

System.out.println("Average Salary = "+employeeSalaryStatistics.getAverage());

System.out.println("Total Salary = "+employeeSalaryStatistics.getSum());

**Who is the oldest employee in the organization? What is his age and which department he belongs to?**

Employee oldestEmployee = employeeList.stream().max(Comparator.comparingInt(Employee::getAge)).get();

System.out.println("Name : "+oldestEmployee.getName());

System.out.println("Age : "+oldestEmployee.getAge());

System.out.println("Department : "+oldestEmployee.getDepartment());

**Compute sum of salaries by department**

// Java:

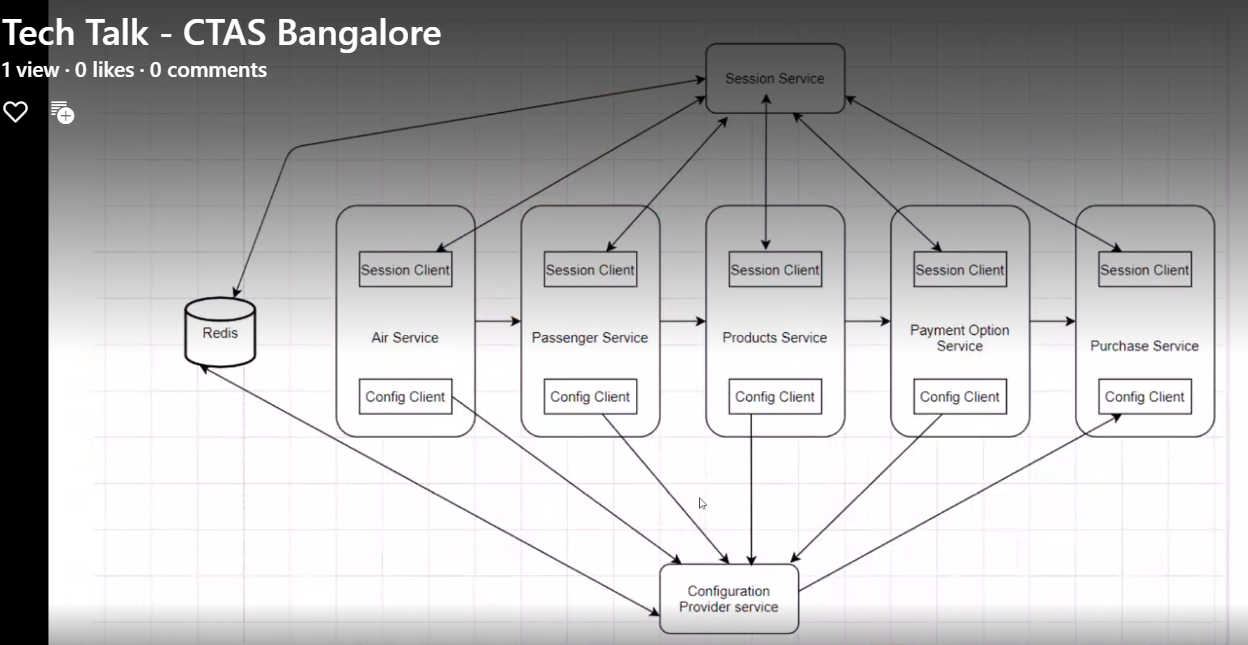
Map<Department, Integer> totalByDept

= employees.stream()

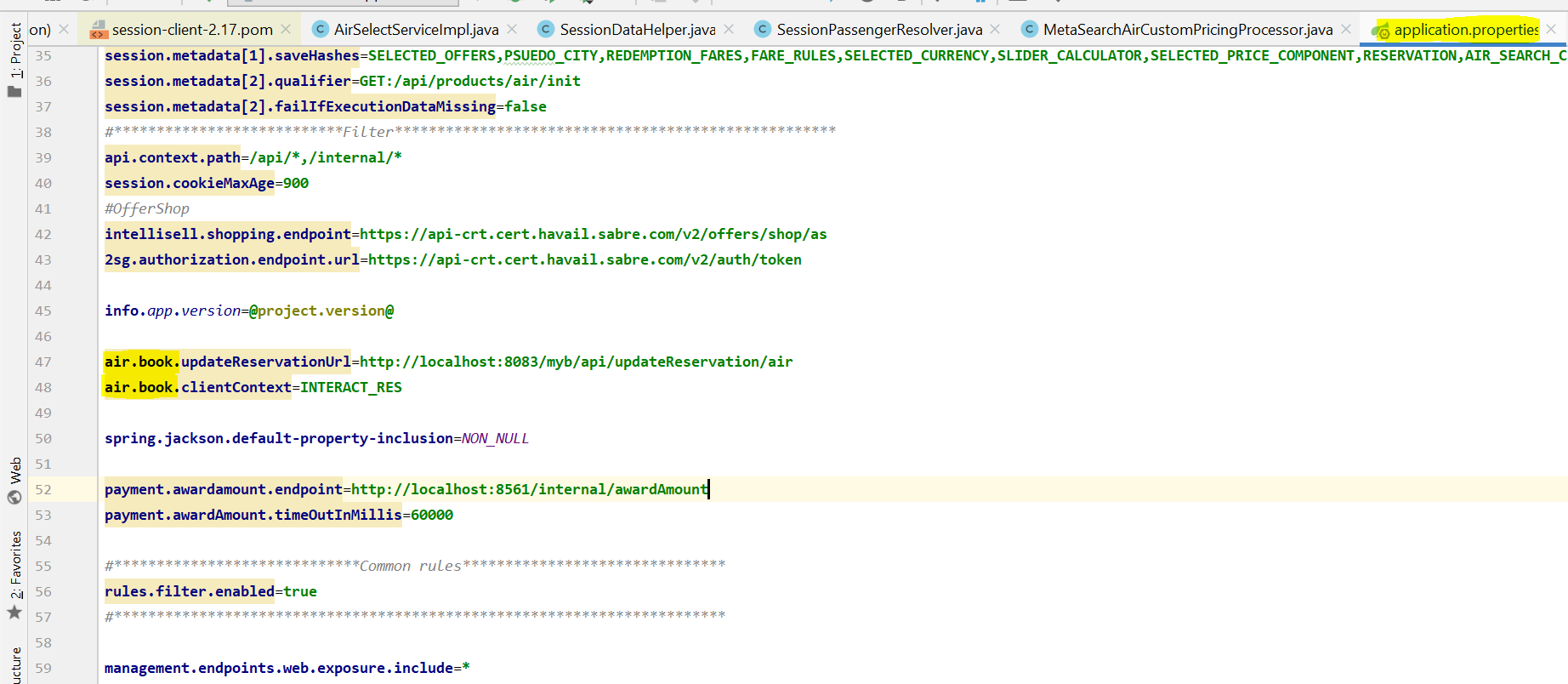
.collect(Collectors.groupingBy(Employee::getDepartment,

Collectors.summingInt(Employee::getSalary)));

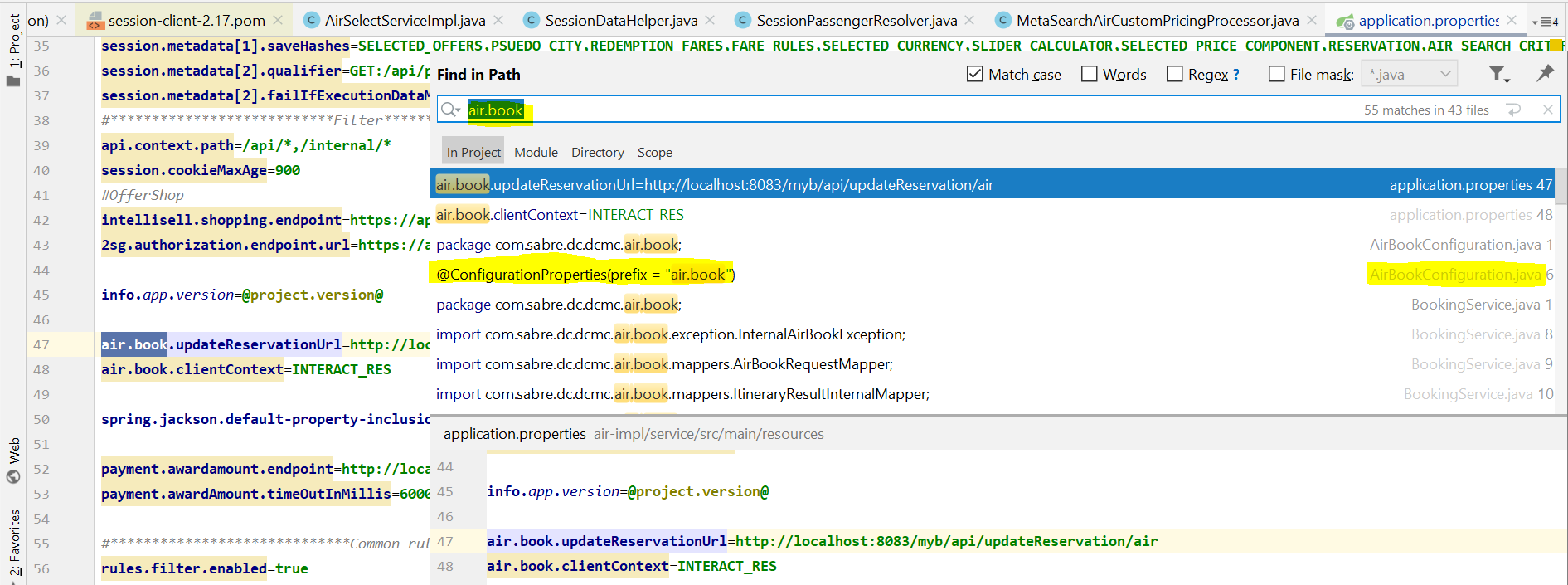
=================================================================================================



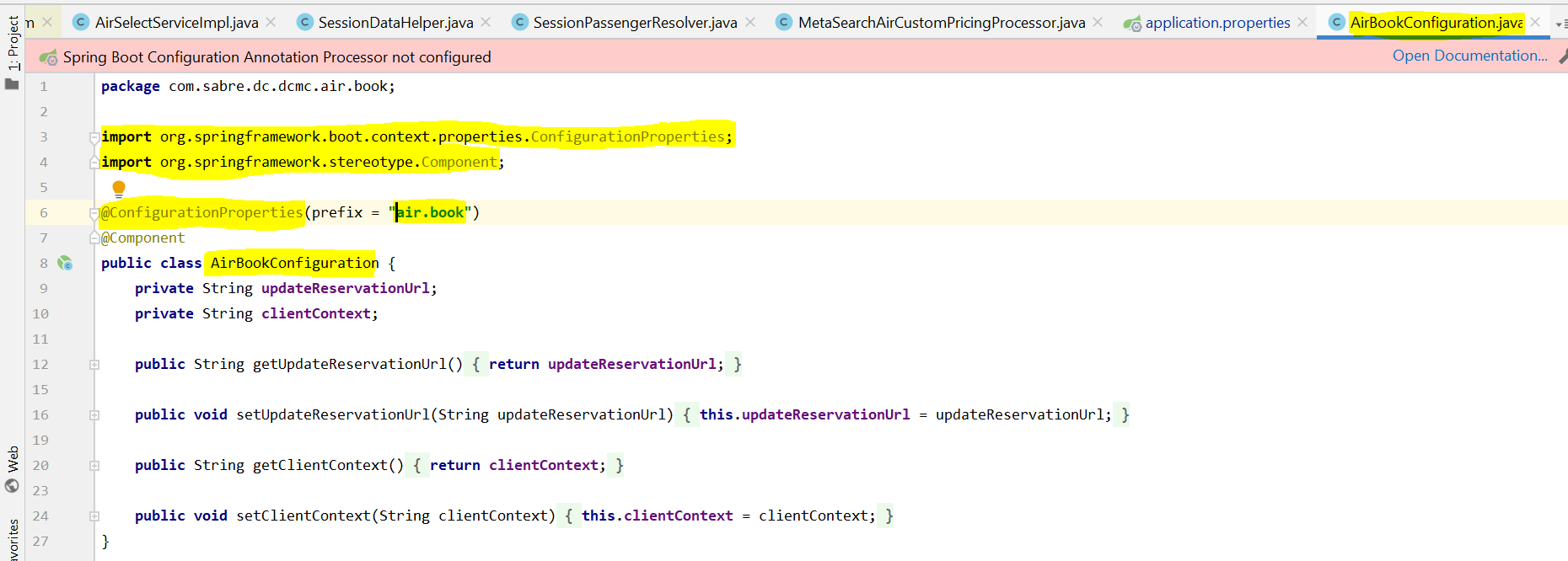
<https://www.tutorialspoint.com/spring_boot/spring_boot_application_properties.htm> 🡪Understanding of application.properties and application.yaml



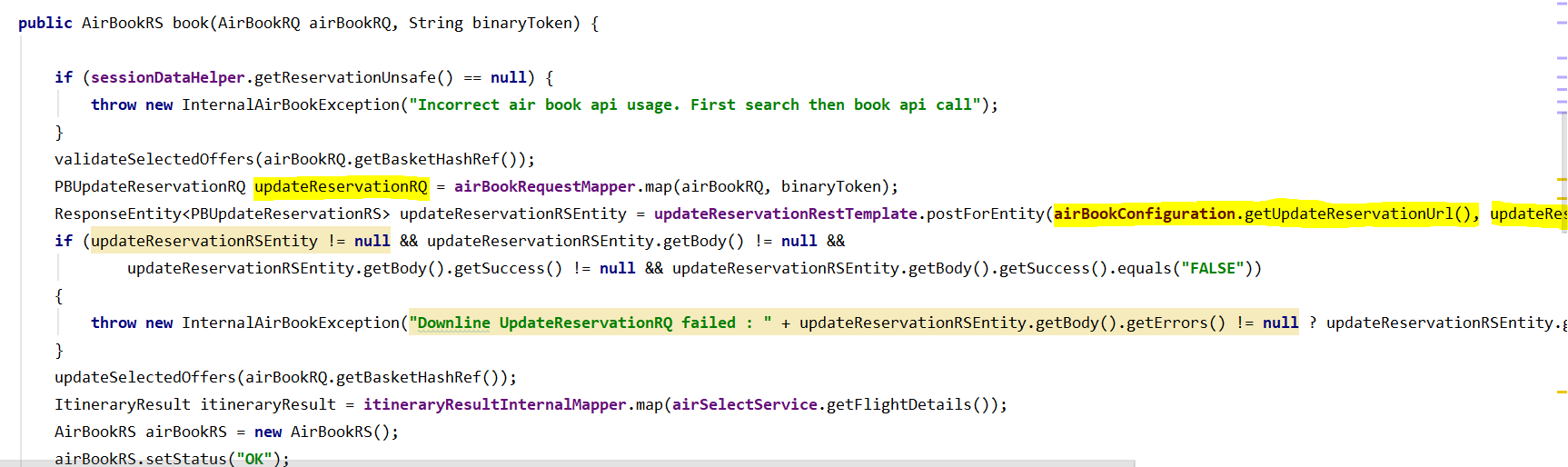
Search air.book

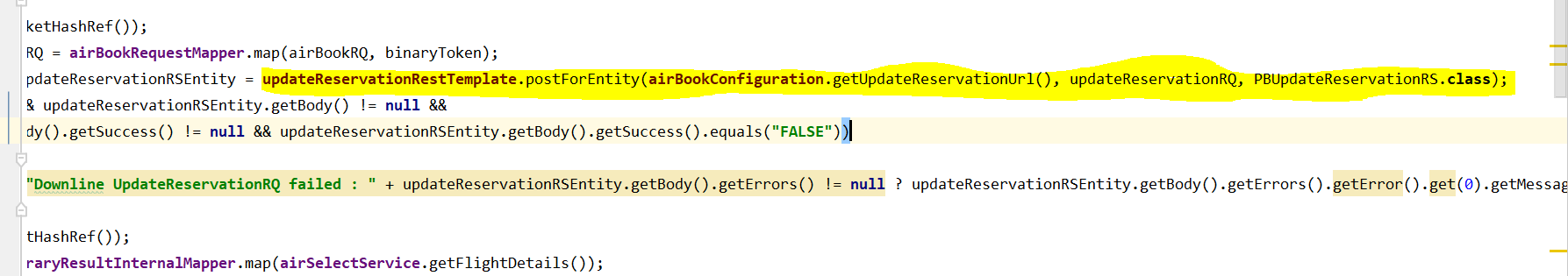


Open **AirBookConfiguration.java**

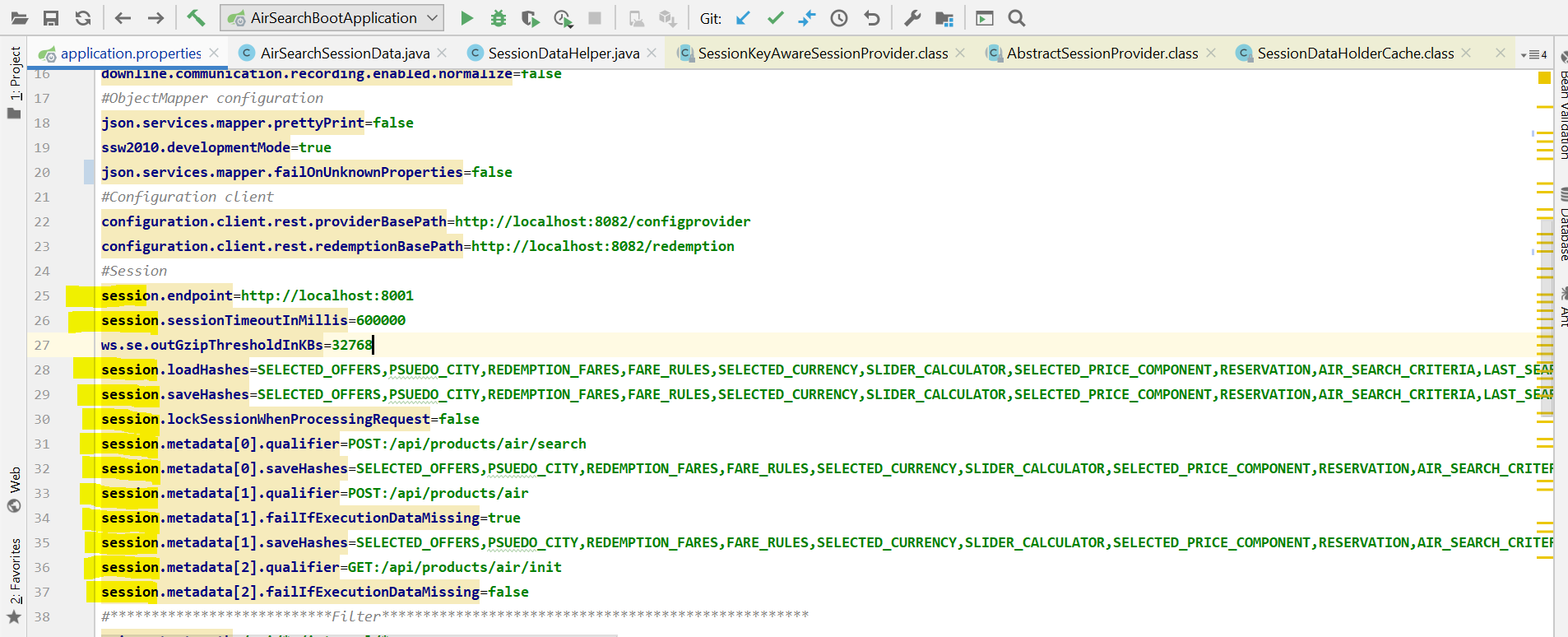








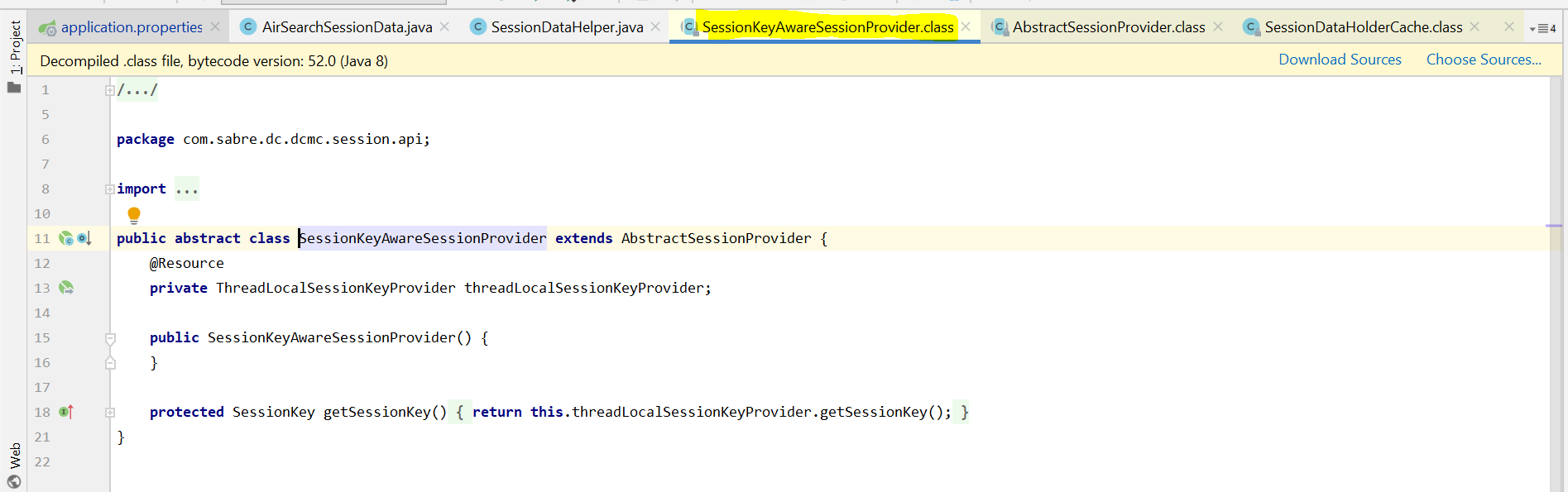
How to Session working:



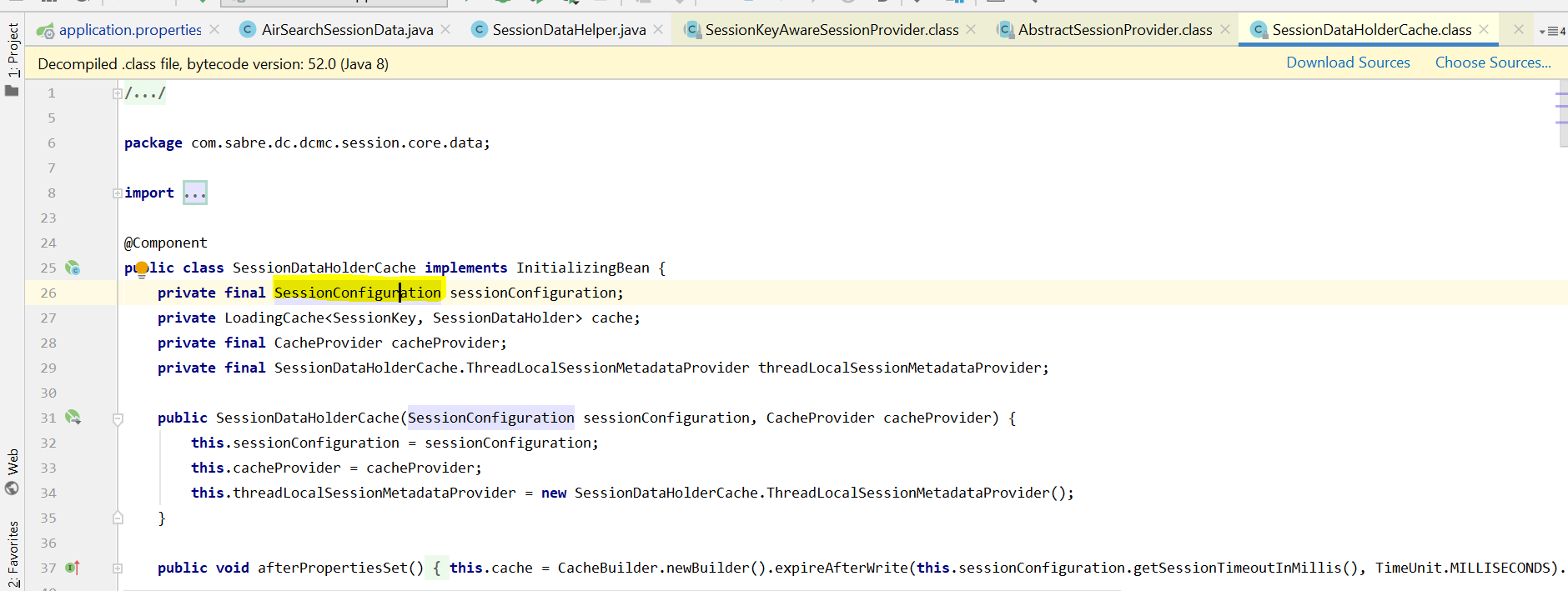


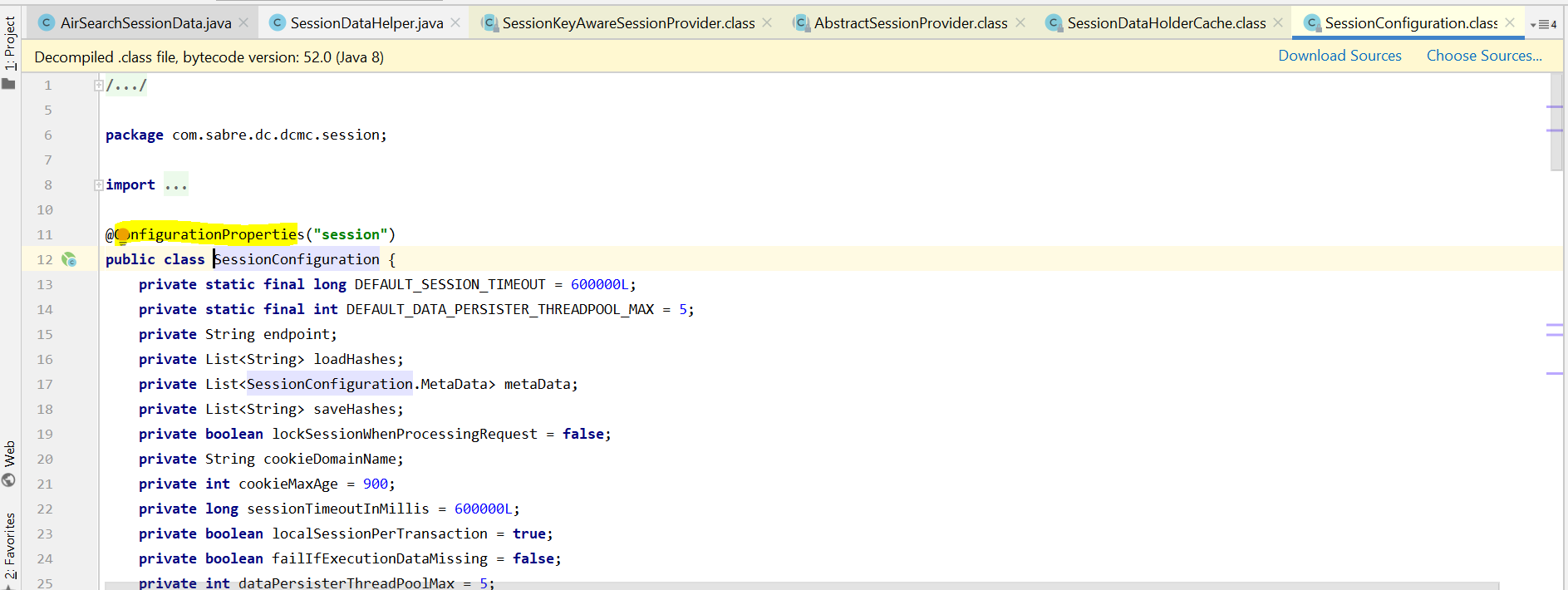


It is invoked session-client jar file







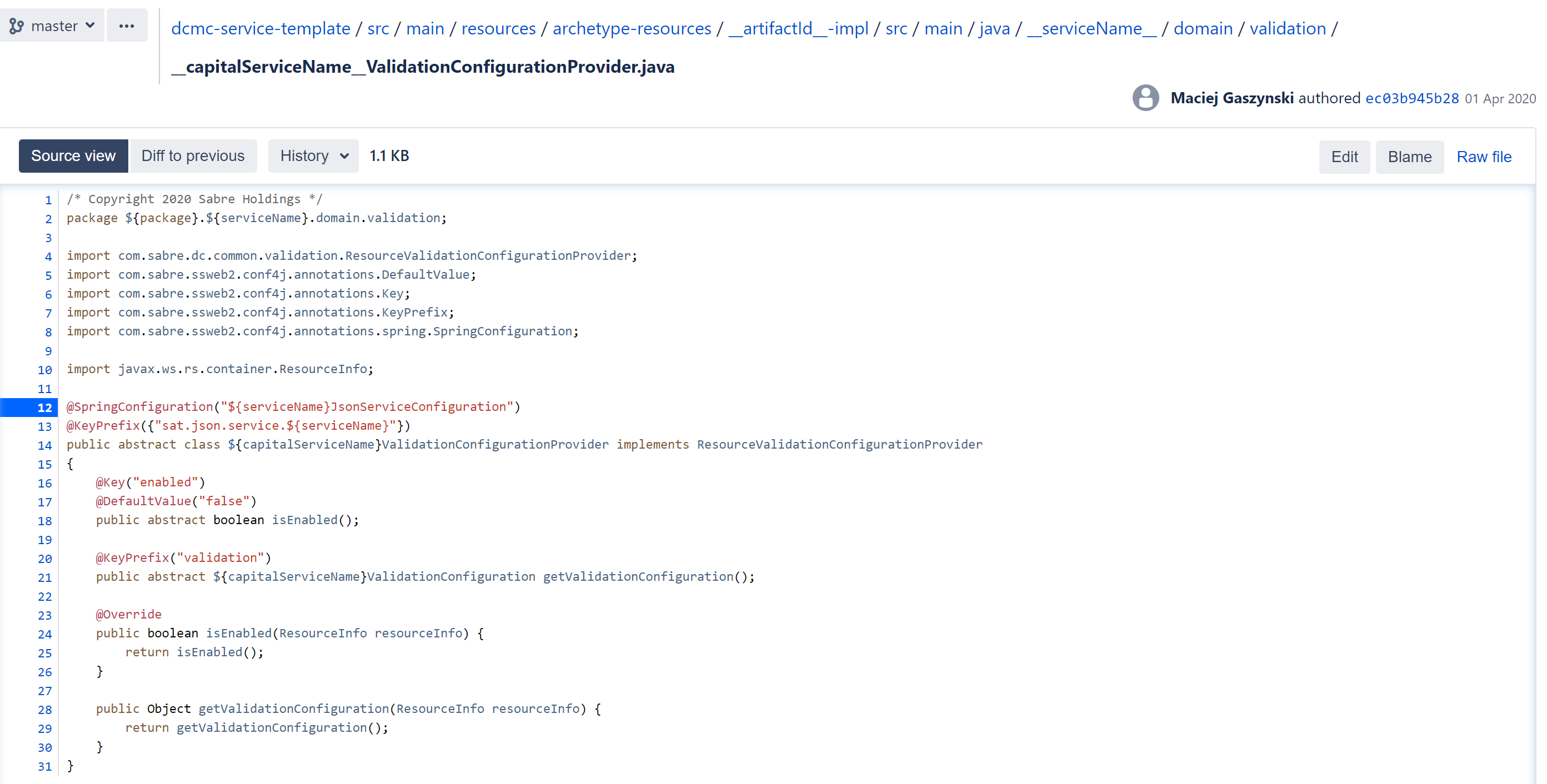


**How to Validation Framework work:**

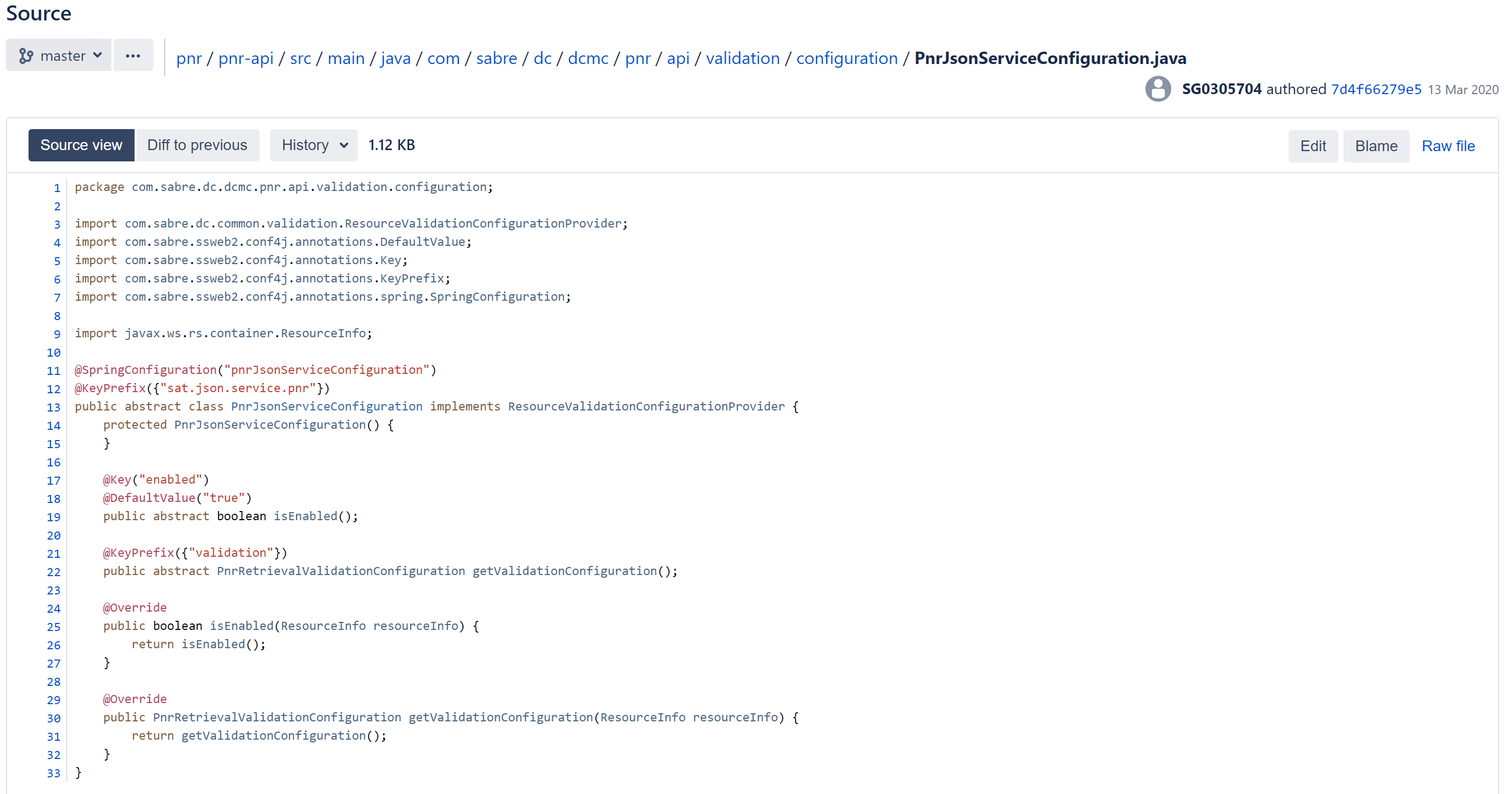
[**https://git.prod.sabre.com/projects/DCMC/repos/shared-lib/browse/conf4j**](https://git.prod.sabre.com/projects/DCMC/repos/shared-lib/browse/conf4j)

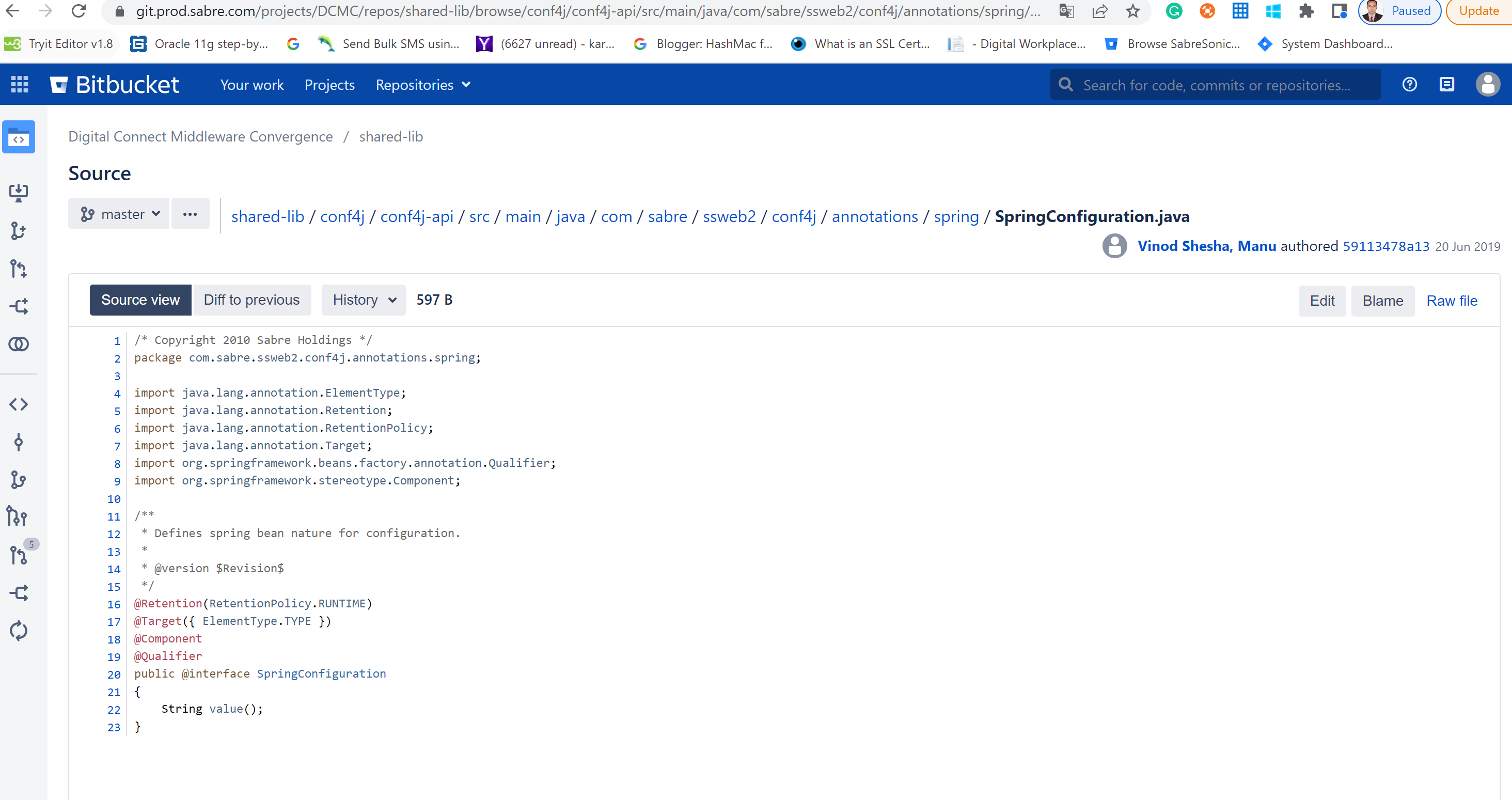
**or**

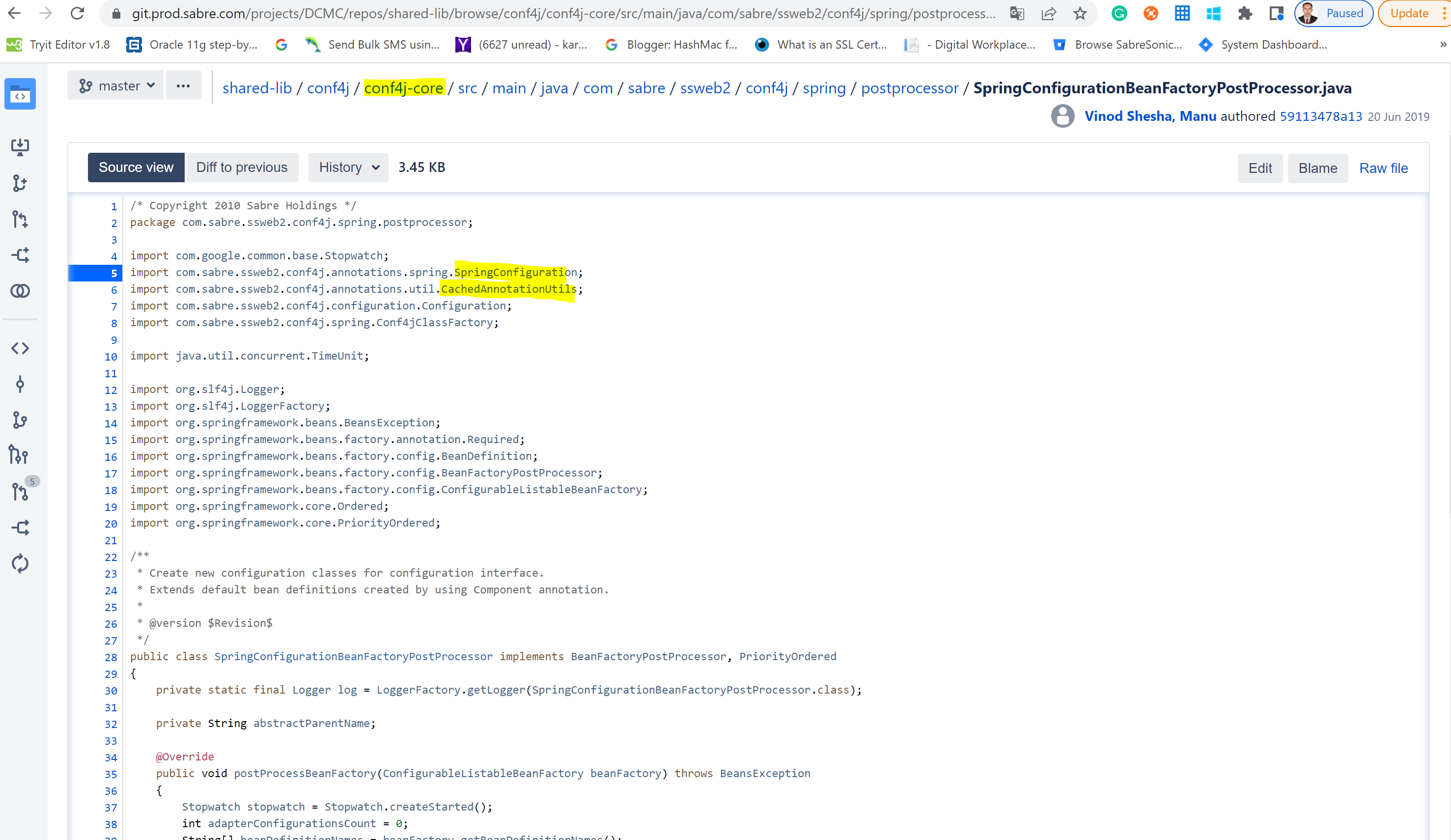
<https://git.prod.sabre.com/projects/DCMC/repos/dcmc-service-template/browse/src/main/resources/archetype-resources/__artifactId__-impl/src/main/java/__serviceName__/domain/validation/__capitalServiceName__ValidationConfigurationProvider.java#12>

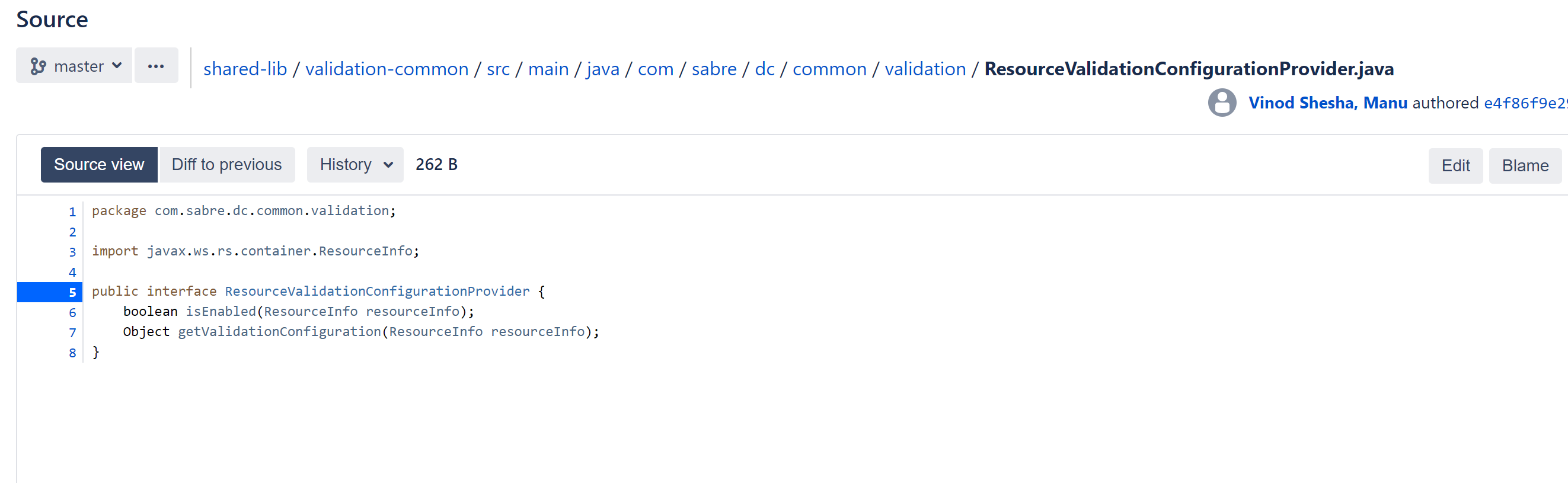


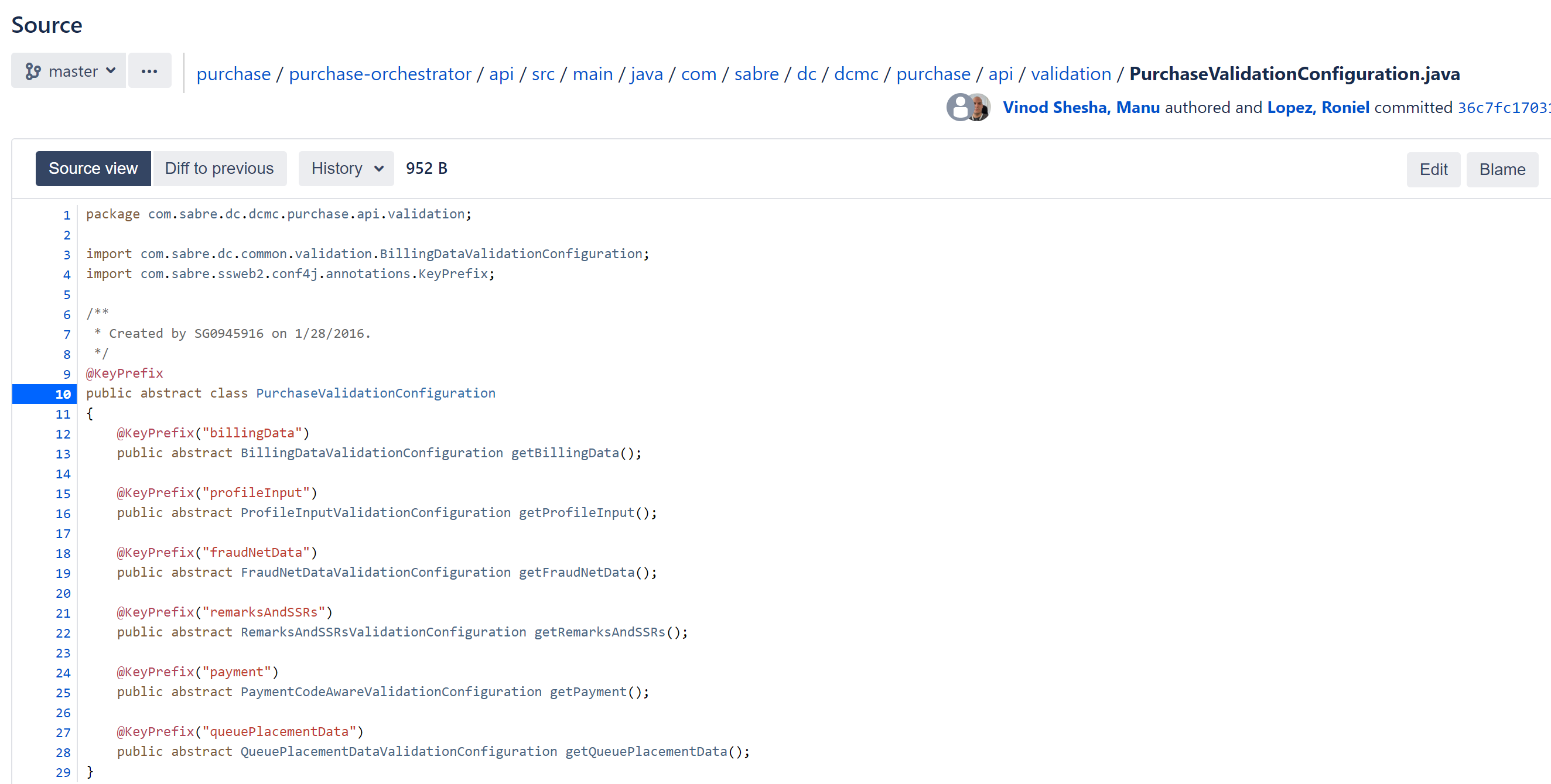
Or other way

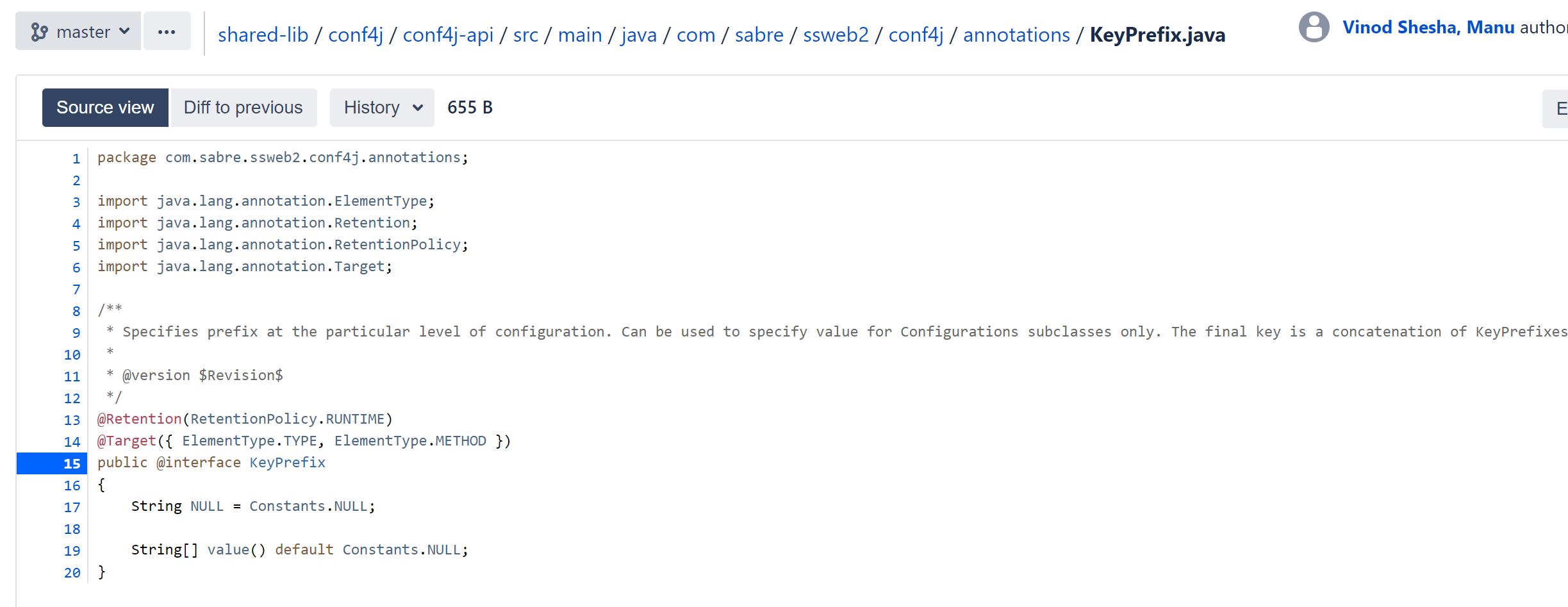






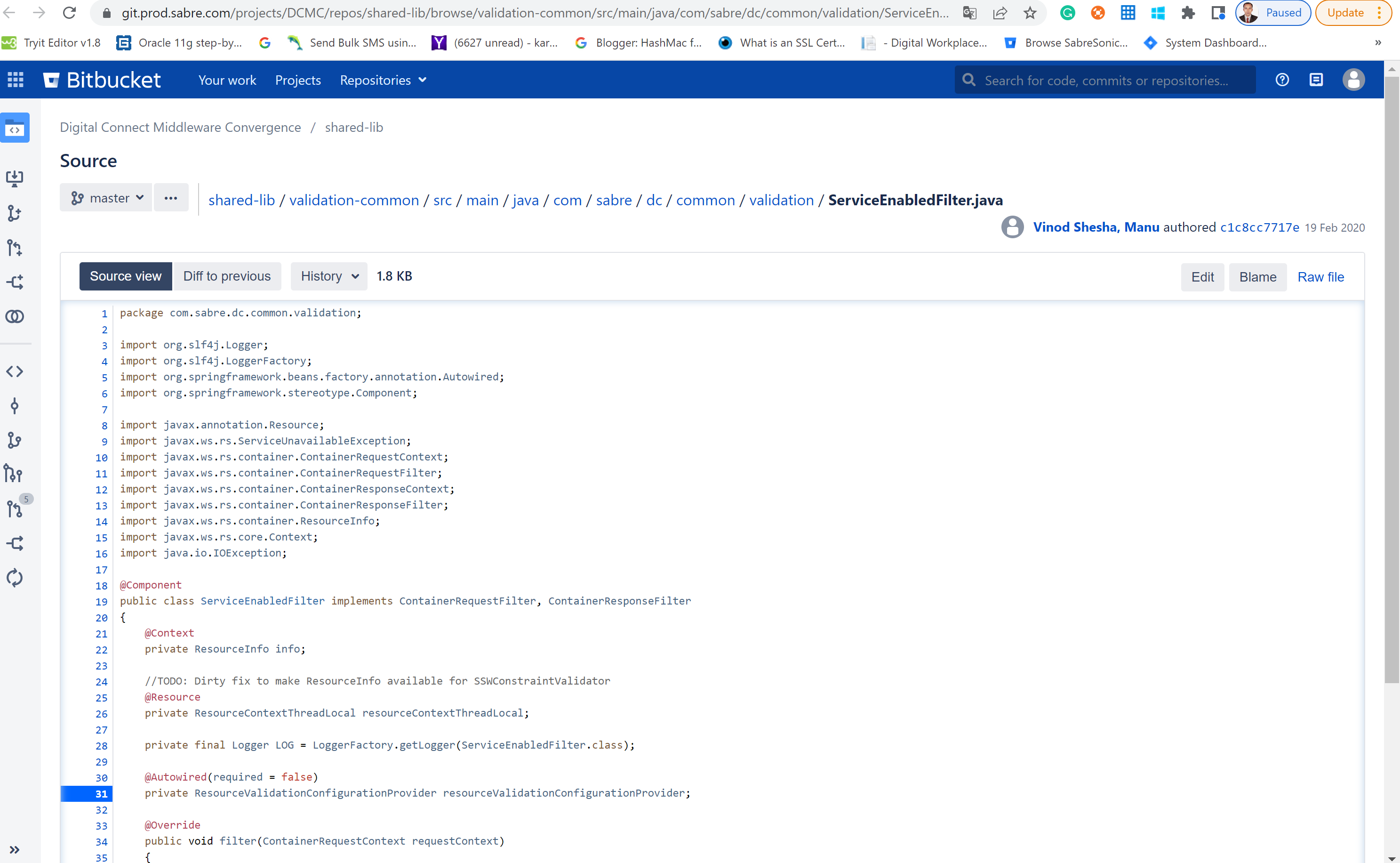


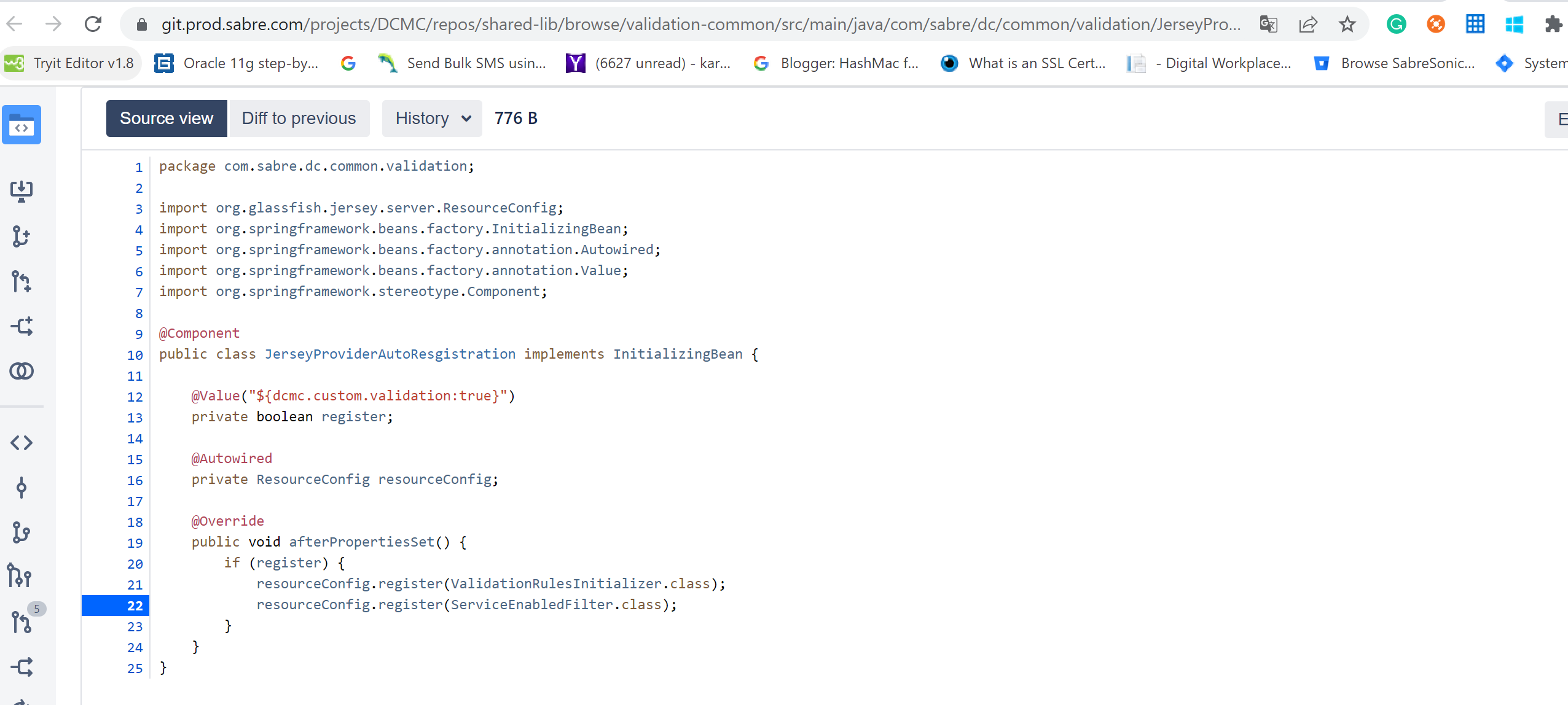




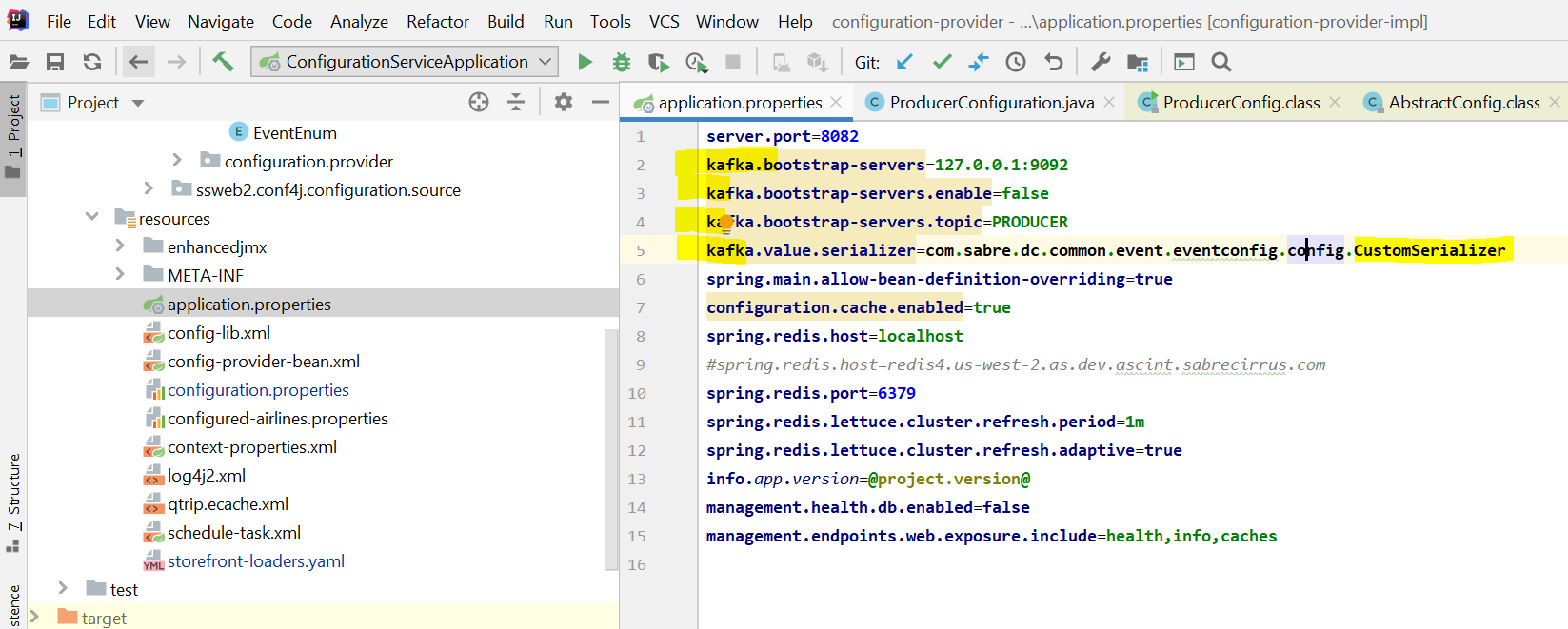


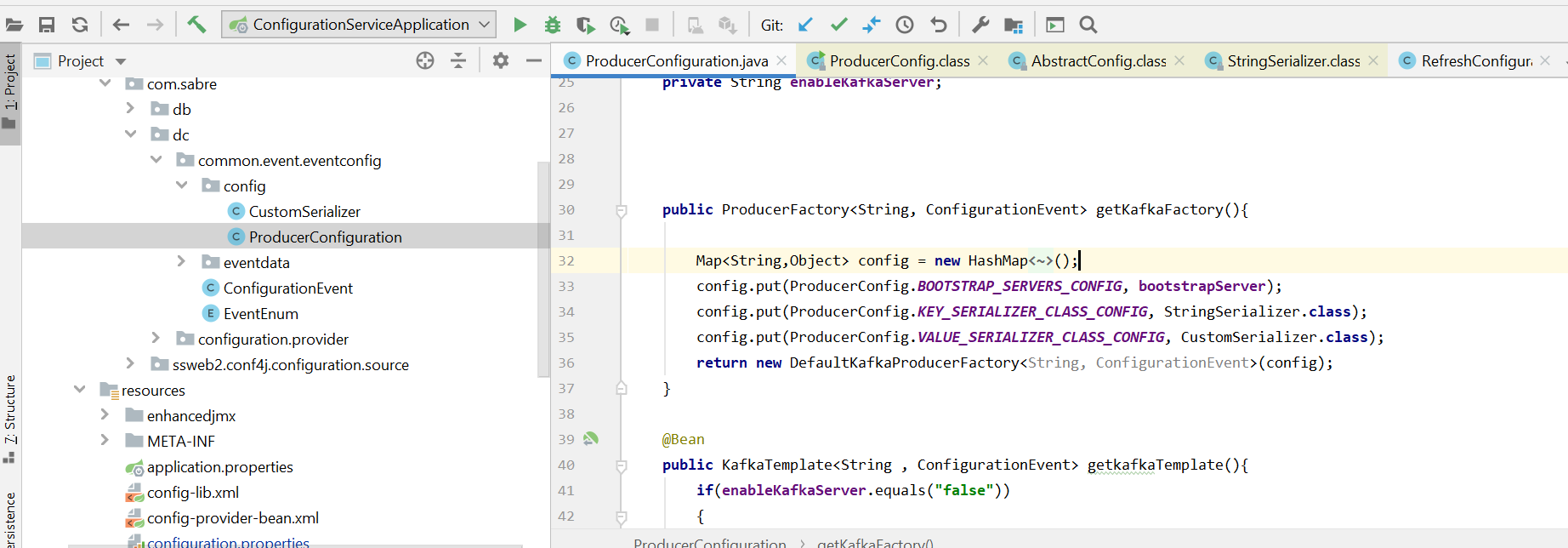




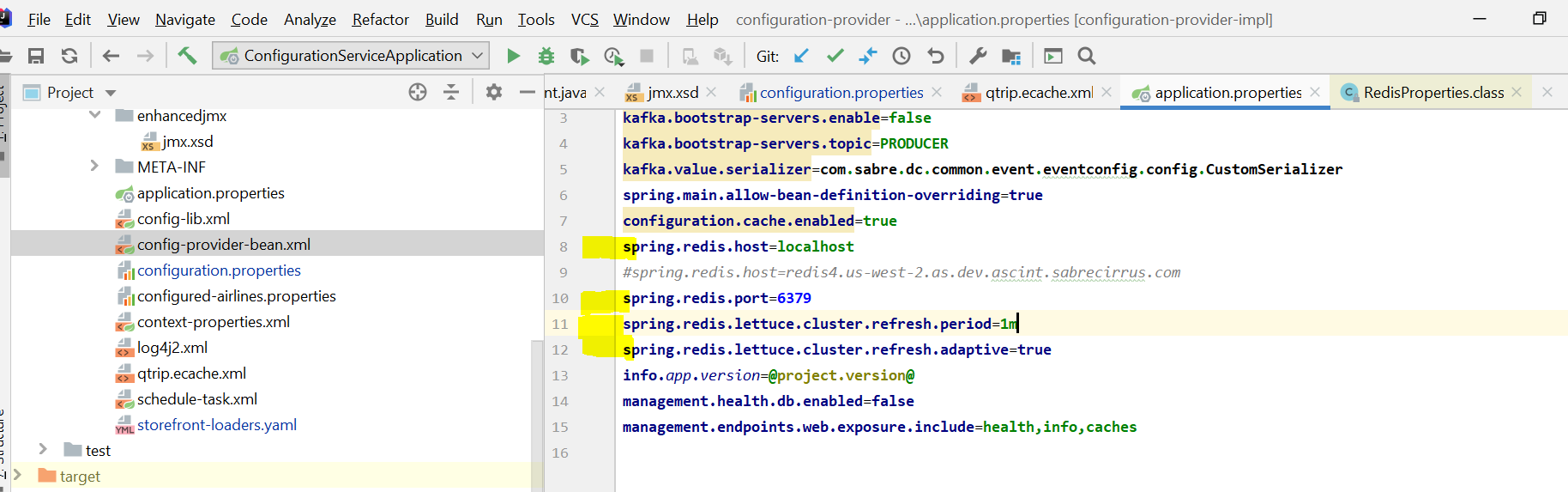


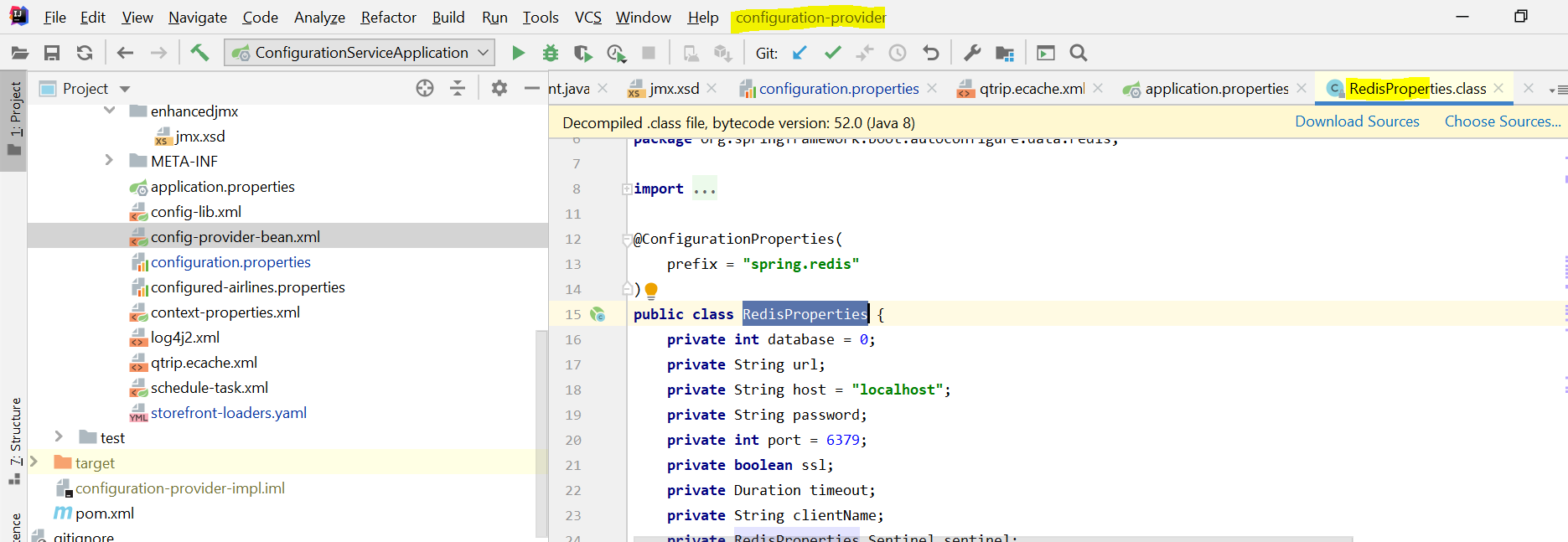
**KAFKA Setup in Configuration-provider module**





Redis:





**USG and rest client everything where it is configured:**

Any application.properties file we open and see the below properties file

Like

*#AWS***usg.ws.endpoint**=**https://sws-crt.cert.havail.kartik.com  
extauth.ws.https.policy.receiveTimeoutInMillis**=**60000  
extauth.ws.https.policy.connectionType**=**KEEP\_ALIVE  
extauth.ws.endpoint**=**https://ppe-eiip.etihad.ae/wsrv/glc/transition/ap/v1.0/EY\_Authenticate  
usg.pool.perAirlineEnabled**=**true  
  
session.loadHashes**=**CONFIG\_VERSION,PROPERTY\_CONFIG\_STRUCT,MUTATION\_CONTAINER,USER\_PROFILE,IS\_LOGGED\_OUT  
session.saveHashes**=**CONFIG\_VERSION,PROPERTY\_CONFIG\_STRUCT,MUTATION\_CONTAINER,USER\_PROFILE,IS\_LOGGED\_OUT***#configuration.client.rest.providerBasePath=http://localhost:8082/configprovider/***configuration.client.rest.providerBasePath**=**http://dcmc1-cp.as.dev.ascint.kartikcirrus.com/configprovider/**

So if you did not found any implementation then definitely it is in **dcmc-shared-lib module** implementation is there.

* USG 🡪usg-lib4j
* Configuration-client 🡪configuration-client

For 2SG geteway using spring cloud gateway Example:

* <https://www.youtube.com/watch?v=Aty83SQD8O4>
* <https://www.youtube.com/watch?v=PfzVE3AOhJ0>

For adapter design pattern we need to look **Orika ja**r file

**For CORS understand:**

CORS 🡪Cross-Origin Resource Sharing: is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources

**Debouncing and Throttling** both Avoiding unnecessary API calls. So it dependents on use case scenarios. When google search we basically used **Debouncing[typing some word and break and some delta time if this delta time is like more than our configuration time then we need to call api otherwise we need to ignore or avoid in middle all call],**

When automatic in a cycle time then we need to used **Throttling.** Example **Machine gun**

