

```

public static CompletableFuture<Void> addOffersForBook(BookEntityReceived bookEntityReceived,
                                                    ConcurrentHashMap<Seller, HashSet<BookResult>> calculatedResult) {
    return callApi(bookEntityReceived).thenCompose(AllegroRequestHandler::parseResponseToOffers)
        .thenApplyAsync(books ->
            IntStream.range(0, books.size() - 1)
                .mapToObj(books::get)
                .collect(Collectors.toList())
        ).thenAcceptAsync(list ->
            list.parallelStream().map(JsonElement::getAsJsonObject).forEach(singleBook -> {
                var seller = extractSellerFromJson(singleBook);
                var newBook = extractBookResultFromJson(singleBook);
                if (seller == null || newBook == null) {
                    AllegroRequestHandler.logger.info("Could not create book or seller from json offer object");
                    return;
                }
                newBook.setBookTitle(bookEntityReceived.title);
                newBook.setWriter(bookEntityReceived.writer);
                calculatedResult.putIfAbsent(seller, new HashSet<>());
                calculatedResult.get(seller).add(newBook);
            }));
}

```