# Marketing Services Specification

[Marketing Services Specification 1](#_Toc813221291)

[Document Overview 1](#_Toc601652303)

[Overview of Data required and provided by the Box Office Team 2](#_Toc143183406)

[Data required by the Box Office Team 2](#_Toc929580409)

[Data to be provided by the Box Office Team 2](#_Toc1336250377)

[Example data fields and methods 3](#_Toc689640428)

[Data required by the Box Office Team 3](#_Toc477925886)

[Data provided by the Box Office Team 5](#_Toc13660788)

[Responsibilities and Communication 6](#_Toc385681618)

# Document Overview

The following document specifies interfaces required to exchange data between the Box Office and Marketing teams. The method of data exchange will be confirmed by the developers, presumably being facilitated through either structured file exports (CSV or JSON) or API endpoints (JDBC).

# Overview of Data required and provided by the Box Office Team

## Data required by the Box Office Team

* **Group booking data**
  + The *Marketing* Team is responsible for bookings for over 12 people. They can hold several seats (rows) while they confirm group bookings. The *Box Office* Team needs to ensure that they don’t list these seats for sale.
    - This needs to be viewable. It will not need to be editable by the *Box Office* Team.
* **“Friends of Lancaster” data**
  + The *Marketing* Team has a list of “Friends of Lancaster”, who have priority access over tickets and seats. The *Box Office* Team needs to have access to this list so they can hold the best seats from sale for a timed period to allow priority to these friends.
    - This needs to be viewable. It will not need to be editable by the *Box Office* Team.
* **Film Data**
  + The *Marketing* Team can add additional activities to the calendar. These tickets are sold by the *Box Office* Team. The Box Office Team require information on the activity to sell tickets.
    - (The *Marketing* Team updates the calendar managed by the *Operations* Team with films, and then the *Box Office* Team accesses this calendar. So, this *could be removed*, as we already will have access to the calendar in another interface)

## Data to be provided by the Box Office Team

* **Ticket sales data**
  + The *Marketing* Team requires details on overall sales insights
  + (This could be split between an online sales API interface and an in-person sales interface)
    - This needs to be viewable, however the *Marketing* Team will not need to edit it.
* **Seating data**
  + The *Marketing* Team needs access to the seats held by “Friends of Lancaster” to send reminders to the group about upcoming shows.
  + The *Marketing* Team needs access to the seats reserved by groups to hold, remove or confirm group bookings.
    - This needs to be viewable, the *Marketing* Team could edit the seats themselves, but more likely the *Box Office* Team (using the group booking data from the *Marketing* Team) will edit this data.
* **Customer Data**
  + The *Marketing* Team needs to access details of the customers to be able to draw insights into sales.
    - This needs to be viewable. It will not need to be editable by the *Marketing* team.

# Example data fields and methods

## Data required by the Box Office Team

**Group Booking Data** (managed by Marketing Team)

* Fields:
  + groupBookingID – int
  + customerID - int
  + eventID – int
  + heldSeats – list<Integer>
  + expiryTime – date
  + status – string (pending, confirmed, cancelled)
* Methods:
  + List<GroupBooking> getHeldSeats(int eventID)
  + void cancelGroupBooking(int groupBookingID, int customerID)
  + void releaseExpiredHoldings()

**Friends of Lancaster Data** (managed by Marketing)

* Fields:
  + friendID – int
  + customerID - int
  + priorityExpiryDate – date
  + heldSeats – list<Integer>
* Methods:
  + Friend getFriend(int )
  + List<Friend> getPriorityHolders()
  + void holdSeatsForFriend(int friendID, int eventID)

**Film Data** (managed by Marketing Team - through Operations Team’s Calendar)

* Fields:
  + filmID – int
  + title – string
  + duration – int
  + description – string
  + rating – string
* Methods:
  + (data comes from methods used for the calendar)

## Data provided by the Box Office Team

**Ticket Sales Data** (provided to Marketing Team)

* Fields:
  + ticketID – int
  + seatID - int
  + eventID – int
  + customerID - int
  + price – double
  + paymentStatus – string (paid, pending, refunded)
  + purchaseDate – date
* Methods:
  + List<TicketSale> getSalesReport(Date startDate, Date endDate)
  + double getRevenueSummary(Date startDate, Date endDate)
  + List<TicketSale> getTicketByCustomer(int customerID)
  + TicketSale getTicketForEvent(int eventID, int customerID)

**Seating Data** (for Friends of Lancaster and Group Bookings)

* Fields:
  + seatID – int
  + eventID – int
  + isHeld – boolean
  + heldByGroupID – int (can be null/empty)
  + heldByFriendID – int (can be null/empty)
* Methods:
  + List<Seat> getHeldSeats()
  + void confirmGroupBooking(int groupBookingID)
  + void releaseGroupBooking(int groupBookingID)

**Customer Data** (provided to Marketing Team)

* Fields:
  + customerID – int
  + fullName – string
  + email – string
  + phoneNumber – string
  + age – int
  + gender - string
  + hasDisability – boolean
  + accessibilityNeeds – string (can be null/empty)
* Methods:
  + Customer getCustomerDetails(int customerID)
  + List<Event> getEventsAttendedByCustomer(int customerID)
  + void updateCustomerInfo(Customer customer)
  + List<Customer> getCustomersWithAccessibilityNeeds()

# Responsibilities and Communication

*Each team is responsible for maintaining their respective APIs and ensuring data accuracy*

*Regular communication/meetings should be facilitated to discuss establish mutual data to work on and any required changes to the interfaces.*