# Operations Services Specification

[Operations Services Specification 1](#_Toc2128653353)

[Document Overview 1](#_Toc1389878751)

[Overview of Data required and provided by the Box Office Team 2](#_Toc1457520849)

[Data required by the Box Office Team 2](#_Toc111204367)

[Data to be provided by the Box Office Team 2](#_Toc879802921)

[Example data fields and methods 3](#_Toc1229278132)

[Data required by the Box Office Team 3](#_Toc1946210051)

[Data provided by the Box Office Team 4](#_Toc700865878)

[Responsibilities and Communication 5](#_Toc388505393)

# Document Overview

The following document specifies interfaces required to exchange data between the Box Office and Operations 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

* **Calendar data**
  + The calendar is managed by the *Operations* Team and must be accessed by the *Box Office* Team to sell tickets
    - This needs to be viewable, and editable to be able to change the number of remaining tickets for the event etc.
* **The various halls seating configurations (Seating data)**
  + The hall's seating configuration is managed by the *Operations* Team and must be reviewed by the Box Office manager/deputy in order to review and remove/discount seats with restricted views, and to make any changes to meet accessibility needs.
    - This needs to be viewable. The customer needs do not suggest that the *Box Office* Team need to edit this data.

## Data to be provided by the Box Office Team

* **Ticket sales data**
  + The *Operations* Team requires details of ticket sales and revenue from the *Box Office* Team for accounting
  + (This could be split between an online sales API interface and an in-person sales interface)
    - This needs to be viewable, however the *Operations* Team will not need to edit it.
* **Seating data**
  + The *Operations* Team needs access to the seats to be updated on restricted-view seats, wheelchair-accessible seating and overall seating information
    - This needs to be viewable, the *Operations* Team could edit the seats themselves, but more likely the *Box Office* Team will edit this data.
* **Customer Data**
  + The *Operations* Team needs to access details of the customers for general use.
    - This needs to be viewable. It will not need to be editable by the *Operations* team.

# Example data fields and methods

## Data required by the Box Office Team

**Calendar Data** (managed by Operations Team)

* Fields:
  + eventID – int
  + eventName – string
  + evendDate - date/int
  + startTime – time/int
  + endTime – time/int
  + remaningTickets – int
* Methods:
  + List<Event> getUpcomingEvents()
  + Event getEventDetails(int eventID)
  + void updateRemaningTickets(int eventId, int newCount)

**Seating Data** (managed by Operations Team)

* Fields:
  + hallID – int
  + seatID – int
  + seatRow – char
  + seatNumber – int
  + seatType – string (normal or priority)
  + isRestrictedView - boolean
  + isAccessible – boolean
* Methods:
  + List<Seat> getSeatingConfiguration(int hallID)
  + void markSeatAsRestricted(int seatID)
  + void updateSeatAccessibility(int seatID, boolean isAccessible)

## Data provided by the Box Office Team

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

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

**Customer Data** (provided to Operations 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.*