**Movie Tickets Booking - Testcases & Payloads**

# Initialize Ledger with custom initial values for number of theatres, screens, shows, number of tickets per show, max Sodas available

## REST API endpoint and Payloads

GET /initledger

Request

{

"theatresCount": 10,

"screensCount": 7,

"showsCount": 5,

"maxTicketPerShow": 150,

"maxSodasAvailable": 250

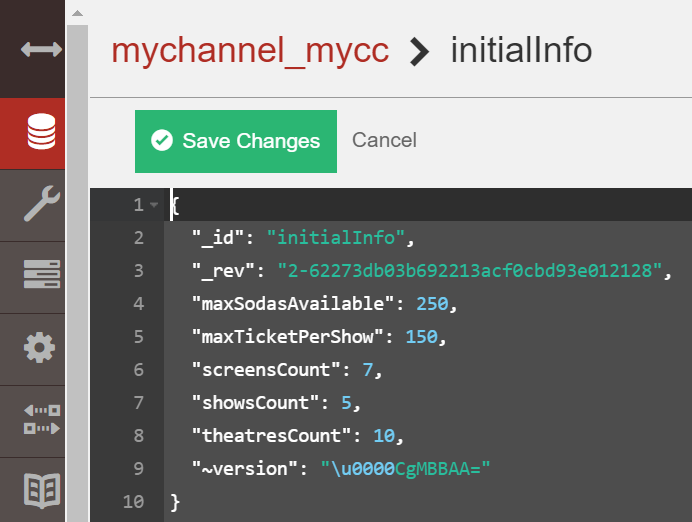
}

Response

“Success” or “Failure”

## Positive testcases

1. Set above mentioned initial parameters with custom values and verify that Ledger has been updated with these values. Can be checked in CouchDB as follows:



## Negative testcases

1. Try with wrong inputs and see that application throw error

# Issue Tickets

## REST API endpoint and Payloads

POST /issuetickets

Request

{

"theatre": "theatre2",

"screen": "screen1",

"show": "show3",

"buyerName": "Konda",

"numberOfTicketsBuying": 20,

"showDate": "2020-12-02"

}

Response

{

"status": "OK",

"statusCode": 200,

"message": "Tickets are issued successfully.",

"ticketInfo": {

"ticketNumber": "**20201202145505513**",

"buyerName": "Konda",

"numberOfTicketsBooked": 20,

"randomNumberForSodaExchange": null,

"exchangedWithSoda": false,

"numberOfSodasExchanged": null,

"showDate": "2020-12-02"

}

}

## Positive testcases

1. Book one ticket.
2. Book more than one ticket at one shot (in a request).
3. Book tickets for future date (not today)
4. Try without providing showDate info, tickets should be booked for today.
5. **Special case**: let’s say, buyer has asked for 10 tickets, but only 5 tickets are available. (in this case, we are not allowing partial bookings)

## Negative testcases

1. Try to book tickets for the show for which tickets are not available
2. Try to book for past date (not implement due to time constaint)

# Get the tickets availability info for given theatre, screen, show and date

## REST API endpoint and Payloads

POST /getnoofavailabletickets

Request

{

"theatre": "theatre2",

"screen": "screen1",

"show": "show3",

"showDate": "2020-12-02"

}

Response

<number of available tickets. 0, if tickets are not available>

## Positive testcases

1. Try for the show for which not even single ticket is booked
2. Try for the show for which some tickets are booked

## Negative testcases

1. Try for the show for which max allowed tickets are booked

# Grab Water bottle and Popcorn

## REST API endpoint and Payloads

PUT /grabwaterbottleandpopcorn

Request

<Ticket number in request body>

Response

{

"status": "OK",

"statusCode": 200,

"message": "Water bottle and Popcorn has been given successfully.",

"ticketInfo": {

"ticketNumber": "20201202145505513",

"buyerName": "Konda",

"numberOfTicketsBooked": 20,

"randomNumberForSodaExchange": **231877564**,

"exchangedWithSoda": **false**,

"numberOfSodasExchanged": **null**,

"showDate": "2020-12-02"

}

}

## Positive testcases

1. Try to grab the water bottle and popcorn

## Negative testcases

1. Try to grab for the ticket which is booked for future date (not today) **(we are allowing if and only if their show date is today)**
2. Try with invalid ticket number

# Exchange Water bottle with Soda

## REST API endpoint and Payloads

PUT /exchangewithsoda

Request

<Ticket number in request body>

Response

{

     "status": "OK",

     "statusCode": 200,

   "message": "Water bottle has been exchanged with Soda successfully.",

    "ticketInfo": {

         "ticketNumber": "20201202145505513",

         "buyerName": "Konda",

         "numberOfTicketsBooked": 20,

         "randomNumberForSodaExchange": 231877564,

         "exchangedWithSoda": **true**,

         "numberOfSodasExchanged": **20**,

         "showDate": "2020-12-02"

}

}

## Positive testcases

1. Try to exchange the water bottle with Soda (for which **even** random number has been generated) – allowed to exchange
2. Try to exchange the water bottle with Soda (for which **odd** random number has been generated) – **NOT allowed to exchange**
3. Special case: let’s say, buyer has asked for 10 tickets, but only 5 tickets are available. (in this case, we are not allowing partial bookings)

## Negative testcases

1. Try to exchange when Sodas are not available
2. Try to grab for the ticket which is booked for future date (not today) **(we are allowing if and only if their show date is today)**
3. Try with invalid ticket number

# Get the Sodas availability info for today

## REST API endpoint and Payloads

GET /getnoofavailablesodas

Request

<nothing>

Response

<number of available tickets. 0, if tickets are not available>

## Positive testcases

1. Try to get number of Sodas remaining for today

## Negative testcases