

BUSES API Document

Version 1.6



Table of Contents

Version Control.....	iv
Escalation Matrix.....	iv
Technical Support.....	iv
1 Overview	1
1.1 Passing Request Data.....	1
1.2 Security (API Authentication)	1
1.3 Compression	1
1.4 Output Formats	2
1.5 Seat Types.....	2
1.6 Trip Types.....	2
1.7 Booking Status	2
1.8 Status and Messages	2
2 Bus Sources – Sources()	4
2.1 Request URL.....	4
2.2 Sample Response.....	4
3 Buses Availability - AvailableBuses().....	5
3.1 Request URL.....	5
3.2 Request Parameters	5
3.3 Response Parameters	5
3.4 Sample Response.....	6
4 Seat Layout - TripDetails().....	9
4.1 Request URL.....	9
4.2 Request Parameters	9
4.3 Response Parameters	9
4.4 Sample Response.....	10
5 Seat Blocking - BlockBusTicket()	13
5.1 Request URL.....	13
5.2 Request Parameters	13
5.3 Sample Request	14
5.4 Response Parameters	15
5.5 Sample Response.....	16
6 Book Seat(s) - BookBusTicket()	18
6.1 Request URL.....	18
6.2 Request Parameters	18

6.3	Response Parameters	18
6.4	Sample Response.....	18
7	Cancellation Details - CancellationDetails()	19
7.1	Request URL.....	19
7.2	Request Parameters	19
7.3	Response Parameters	19
7.4	Sample Response.....	19
8	Cancel Bus Ticket - CancelBusTicket().....	21
8.1	Request URL.....	21
8.2	Request Parameters	21
8.3	Response Parameters	21
8.4	Sample Response.....	21
9	BusTicketBookingDetails()	23
9.1	Request URL.....	23
9.2	Request Parameters	23
9.3	Sample Response.....	23
10	BoardingPointDetails().....	26
10.1	Request for BoardingPointDetails	26
10.2	Request Parameters	26
10.3	Response.....	26
11	Check API Balance - GetAPIBalance()	27
11.1	Request	27
11.2	Response.....	27
12	RTCs Implementation	28
12.1	Overview.....	28
12.2	UPSRTC	29
12.3	HRTC	29
12.4	JKSRTC.....	31
12.5	OSRTC	32
12.6	PEPSU.....	33
13	Certification & Guidelines.....	35
13.1	Certification	35
13.2	[Case 1]: One-way single seat booking.....	35
13.3	[Case 2]: One-way multiple seats booking	35
13.4	[Case 3]: Round-trip single seat booking.....	35
13.5	[Case 4]: Round-trip multiple seats booking	36

13.6 [Case 5]: Cancellation reference numbers	36
14 FAQ(s).....	37
14.1 MISC.....	37
14.2 Sources()	38
14.3 AvailableBuses().....	38
14.4 TripDetails()	39
14.5 BlockBusTicket ().....	40
14.6 BookBusTicket()	41
14.7 CancelBusTicket().....	42
APPENDIX - Error Codes	43

Version Control

Version	Release Date	Modified Date	Operations
V1.6	13 th Jun, 2017	21 st NOV, 2017	RTCs implementation, GetApiBalance() method implemented
V1.5	29 th Nov, 2016		SeatCode parameter updated
V1.4	7 th Oct, 2016		Display Name
V1.3		27 th Aug, 2015	Convenience fee, Service charge
V1.2		20 th Sep, 2014	
V1.1		8 th Jul, 2014	Service Tax, parameters changes
V1.0	1 st Jan, 2014	1 st Jul, 2014	Bus booking Method has changed

Escalation Matrix

Customer Support			
Level	Email ID	Customer Support	TAT for next level escalation
Level 1	BA@i2space.com	040-64626301/03	09:30 AM - 08:00 PM
Level 2	krishna.m@i2space.com	8886216947 / 9505902898	11:30 AM - 09:00 PM
Technical Support			
Level	Email ID	Customer Support	TAT for next level escalation
Level 1	technicalsupport@i2space.com	040-64626301/03	09:30 AM - 05:30 PM
Level 2	technicalsupport@i2space.com	8886216947 / 9505902898	09:30 AM - 05:30 PM
Business			
Level	Email ID	Customer Support	TAT for next level escalation
Level 1	BA@i2space.com	040-64626301/03	09:30 AM - 08:00 PM
Level 2	Partners@i2space.com	8886216947 / 9505902898	11:30 AM - 09:00 PM

Technical Support

For any support during integration and production, our renowned technical team will be at your disbursal from Mon to Fri from **09:30 AM to 05:30 PM IST (MON to FRI excepting Public Holidays)**. Beyond which you can always reach us at technicalsupport@i2space.com.

NOTE If you may face any issue with API, we request you to kindly share complete JSON request and response to resolve the issue in quick and efficient way

1 Overview

1.1 Passing Request Data

Request data is passed to the API by Get and POST JSON objects to the API endpoints with the appropriate parameters. The documentation for each API call will contain more detail on the parameters accepted by the call. As an alternative, you can also use HTTP Get and POST parameters, just like submitting an HTML FORM, but JSON or XML objects are recommended.

1.2 Security (API Authentication)

All users must provide their "**ConsumerKey**" and "**ConsumerSecret**" in every API request. This key is specific to an environment. Therefore, your API Keys will be different for Development and Production environments. Make sure you are using the correct Consumer Key and ConsumerSecret pair.

For REST service calls, the Key should be passed in as an HTTP header element called "ConsumerKey", "ConsumerSecret" and the provided sample code contains examples in .Net of how to use the API Key header.

Passing API Key as a HTTP Header

If you feel that passing API Key in URL parameter is not secure then you can pass the API key as a HTTP Header for every API call

Header Key	Value
ConsumerKey	4638XXXXXXXXXXXXXXXXXXXXE3E8
ConsumerSecret	506CXXXXXXXXXXXXXXXXXXXX55CV

1.3 Compression

For some destinations, you may get around 800 to 1000 results. Transmission of such big responses takes extra time over the internet. To overcome this problem and reduce transmission/ response time by about 80%, system is enabled with gzip compression. Please check the below steps on how to receive compressed response.

Step 1: Request must contain the header as "Accept-Encoding: gzip, deflate".

Step 2: If server detects, the response will be compressed.

Step 3: Client receives response with the header "Content-encoding: gzip".

Step 4: Client needs to decompress the Response before processing.

Header Key	Value
Accept-Encoding	gzip

NOTE Gzip compression is **Optional** (In future it may be mandatory). Without gzip headers you will be able to access all the services in TEST/LIVE environment.

1.4 Output Formats

We support only JSON as output format. The response data for each call will be encoded in the JSON format. Inputs accepted in JSON objects for HTTP POST parameters.

1.5 Seat Types

Bus Types	Description
1	AC Semi-Sleeper
2	AC Sleeper
3	Non-AC Semi-Sleeper
4	Non-AC Sleeper

1.6 Trip Types

Trip Type	Description
1	One Way
2	Round Trip

1.7 Booking Status

Booking Status	Description
1	Ticket Blocked Successfully
2	Ticket Blocking Failed
3	Ticket Booked Successfully
4	Ticket Booking Failed
5	Ticket Cancelled Successfully
6	Cancellation Failed
7	Partial Cancellation Done Successfully
8	Partial Cancellation Failed
9	Failed

1.8 Status and Messages

Code	Message	Description
200	Success	This message indicates Successful completion of a request
400	Not a valid request	One or more parameters for the request has been missed out
401	Authentication failed	API Key is blank or not valid
401/403	Access Denied / Unauthorized	If the request Content Type is NOT application/JSON. Unauthorized when consumer key and consumer key are not Matching

500	Internal Server Error	Errors, Exceptions if any occurs in Server Side while processing a request
-----	-----------------------	--

.

2 Bus Sources – Sources()

2.1 Request URL

Method	URL
GET	Buses/Sources

<http://webapi.i2space.co.in/Buses/Sources>

2.2 Sample Response

```
[
  {
    "Id": 100,
    "Name": "Hyderabad"
  },
  {
    "Id": 101,
    "Name": "Delhi"
  },
  {
    "Id": 102,
    "Name": "Avinashi"
  },
  {
    "Id": 103,
    "Name": "Chennai"
  }
]
```

3 Buses Availability - AvailableBuses()

3.1 Request URL

This API method provides Available Buses in particular route for particular date.

Method	URL
GET	Buses/AvailableBuses?sourceId=<sourceId>&destinationId=<destinationId>&journeyDate=<journeyDate>&tripType=<tripType>&userType=<userType>&user=<user>&returnDate=<returnDate>

```
http://webapi.i2space.co.in/Buses/AvailableBuses?sourceId=103&destinationId=109&journeyDate=11-11-2017&tripType=1&userType=5&user=&returnDate=
```

3.2 Request Parameters

Parameter	Type	Required	Description
sourceId	string	REQUIRED	Source City Id. E.g. 100 (for Hyderabad)
destinationId	string	REQUIRED	Destination City Id. E.g. 109 (for Bangalore)
journeyDate	string	REQUIRED	Travel Journey Date. E.g. 01-11-2017
tripType	string	REQUIRED	Refer TripTypes table. 1
userType	integer	REQUIRED	User Type. It is always 5 (STATIC). In XML request pass as Guest.
returnDate	string	CONDITIONAL	Travel Return Date. Mandatory when it is Round-Trip

3.3 Response Parameters

Parameter	Type	Description
AffiliateId	string	Not Applicable
ArrivalTime	string	Arrival time of bus. 05:30 AM
AvailableSeats	string	No.of Seats available in bus. 41
BoardingTimes	object	BoardingTimesDetails. []
BusType	string	Bus type. Seater AC Volvo Multi Axle Semi Sleeper (2+2)
CancellationPolicy	string	Cancellation charge for bus. 24:-1:20:0;0:48:10:0
ConvenienceFee	string	It is always zero for API partners. 0.00
DepartureTime	string	Departure time of particular bus. 07:30 PM
DestinationId	string	DestinationId of the bus109
DisplayName	string	Bus Name. I2space Tours and Travels

DroppingTimes	object	DroppingTimes Details. []
Duration	string	Journey time 10:00 hrs
Fares	string	Price of the bus. 1200
Id	string	Id of bus is unique. 1038
idProofRequired	boolean	Shows Id proof required or not. false
IsRtc	boolean	It is True if it is RTC bus. false
JourneyDate	string	Date of Journey. 2017-11-01
Mticket	string	If true, then Operator accepts SMS ticket. true
NetFares	string	Price of the bus. 1200
OperatorServiceCharge	string	OperatorServiceCharge of bus. 0.00
ParticalCancellationAllowed	string	If true Partial cancellation allowed else not allowed. true
Provider	string	Operator code. +rfVawweNEABIDWJVZMKFA==
SeatType	string	Refer SeatTypes table. 1
ServiceTax	string	ServiceTax of bus. 0.00
SourcelId	string	Source Id of bus. 100
Travels	string	Name of the travel. I2space Travels

BoardingTimes[]/DroppingTimes[]:

Parameter	Type	Description
Address	string	Address. null
ContactNumbers	string	Contact Numbers. null
ContactPersons	string	Contact Persons. null
PointId	string	Point Id. 2500
Landmark	string	Landmark. null
Location	string	Location. Chennai central
Name	string	Name. null
Time	string	Time in minutes. 960

3.4 Sample Response

```
{
  "AvailableTrips": [
```

```
{
  "DisplayName": "GDS Demo Test",
  "AvailableSeats": "28",
  "IsRtc": false,
  "ArrivalTime": "02:00 AM",
  "BoardingTimes": [
    {
      "Address": null,
      "ContactNumbers": null,
      "ContactPersons": null,
      "PointId": "25000",
      "Landmark": null,
      "Location": "Chennai central",
      "Name": null,
      "Time": "960"
    }
  ],
  "BusType": "NON/AC , SLEEPER",
  "CancellationPolicy": "0:-0:100;6:-0:40;7:-0:30;8:-0:20;9:-0:0;10:-0:0",
  "idProofRequired": false,
  "DepartureTime": "04:00 PM",
  "DestinationId": "4292",
  "Duration": "10:00 hrs",
  "DroppingTimes": [
    {
      "Address": null,
      "ContactNumbers": null,
      "ContactPersons": null,
      "PointId": "45176",
      "Landmark": null,
      "Location": "Koramangala",
      "Name": null,
      "Time": "120"
    },
    {
      "Address": null,
      "ContactNumbers": null,
      "ContactPersons": null,
      "PointId": "1385",
      "Landmark": null,
      "Location": "Majestic",
      "Name": null,
      "Time": "130"
    }
  ],
  "Fares": "102",
  "ServiceTax": "0",
  "OperatorServiceCharge": "0.00",
  "ConvenienceFee": 0,
  "ConvenienceFeeType": 1,
  "AffiliateId": null,
  "PartnerFareDatails": {
    "NetFares": "102",
    "Commission": "12",
    "CommissionType": 0
  },
  "NetFares": null,
  "OperatorId": "3963",
  "Id": "85",
  "Provider": "DZ/GmxpuN0rTfFNIdYpjWA==",
  "PartialCancellationAllowed": "false",
  "SeatType": 4,
}
```

```
    "SourceId": "4562",  
    "Travels": "GDS Demo Test",  
    "Mticket": "true",  
    "InventoryType": null,  
    "Journeydate": "11-11-2017"  
  }  
],  
"ProvidersCount": 2,  
"ResponseStatus": 200,  
"Message": "Sucess"  
}
```

NOTE You'll get **BoardingTimes**, **DroppingTimes** values either in `AvailableBuses()` or `TripDetails()` method. Implement your coding accordingly in both methods to handle those parameters.

4 Seat Layout - TripDetails()

4.1 Request URL

This API method is used to get the seat layout for particular bus.

Method	URL
GET	Buses/TripDetails?tripId=<tripId>&sourceId=<sourceId>&destinationId=<destinationId>&journeyDate=<journeyDate>&tripType=<tripType>&userType=<userType>&provider=<provider>&travelOperator=<travelOperator>&user=<user>&returnDate=<returnDate>

```
http://webapi.i2space.co.in/Buses/TripDetails?tripId=86&sourceId=103&destinationId=109&journeyDate=11-11-2017&tripType=1&userType=5&provider=DZ/GmxpuN0rTfFNidYpjWA==&travelOperator=GDS Demo Test&user=&returnDate=
```

4.2 Request Parameters

Parameter	Type	Required	Description
tripId	string	REQUIRED	Id parameter of particular bus which is available in bus availability method. 86
sourceId	string	REQUIRED	Id of particular source. 103
destinationId	string	REQUIRED	Id of particular destination. 109
journeyDate	string	REQUIRED	Date of journey. 11-11-2017 (dd-mm-yyyy)
tripType	string	REQUIRED	Refer TripTypes table. 1
userType	integer	REQUIRED	User type. Always 5 (STATIC). In XML request pass as Guest.
provider	string	REQUIRED	Provider parameter value of availableBuses response. DZ/GmxpuN0rTfFNidYpjWA==
travelOperator	string	REQUIRED	Travels parameter of particular bus which is available in AvailableBuses method. GDS Demo Test
user	string	REQUIRED	Always empty.
returnDate	string	CONDITIONAL	Provide when you are searching for roundtrip.

4.3 Response Parameters

Parameter	Type	Description
TripId	string	Id of bus. 86
Seats	array	Contain seat details. []
BoardingTimes	string	BoardingTimes. null
DroppingTimes	string	DroppingTimes. null

Provider	string	Provider of bus. DZ/GmxpuN0rTfFNIdYpjWA==
ResponseStatus	string	Refer Status table. 200
Message	string	Message. SUCCESS

Seats[]:

Parameter	Type	Description
Column, row	string	Defines the position of the seat. 1, 1
Fare	string	Price of particular seat. 102
IsAvailableSeat	string	If it is true then seat is available, else seat is booked. true
IsLadiesSeat	string	If it is true then particular seat is reserved for ladies. false
Length	string	Seater (Length=1) & Sleeper (Length=2). 1
NetFare	string	Price of particular seat (internal purpose). 102
Number	string	It defines seat number. 1
OperatorServiceCharge	string	Operator Service Charge for particular seat. 0
SeatCode	string	Seat code for particular seat. 1S1
Servicetax	string	Service tax for particular seat. 0
Zindex	string	Defines lower berth or upper berth. Lower (z=0) & Upper (z=1). 0

4.4 Sample Response

```
{
  "TripId": "86",
  "Seats": [
    {
      "Column": 0,
      "Fare": "102",
      "NetFare": "102",
      "IsAvailableSeat": "True",
      "IsLadiesSeat": "False",
      "Length": 1,
      "Number": "1",
      "Row": 4,
      "Width": 1,
      "Zindex": 0,
      "Servicetax": "0",
      "OperatorServiceCharge": "0",
      "SeatCode": "0S1"
    },
    {
      "Column": 0,
      "Fare": "102",
      "NetFare": "102",
      "IsAvailableSeat": "True",
      "IsLadiesSeat": "False",

```

```

    "Length": 1,
    "Number": "2",
    "Row": 2,
    "Width": 1,
    "Zindex": 0,
    "Servicetax": "0",
    "OperatorServiceCharge": "0",
    "SeatCode": "1S1"
  },
  {
    "Column": 0,
    "Fare": "102",
    "NetFare": "102",
    "IsAvailableSeat": "True",
    "IsLadiesSeat": "False",
    "Length": 1,
    "Number": "3",
    "Row": 1,
    "Width": 1,
    "Zindex": 0,
    "Servicetax": "0",
    "OperatorServiceCharge": "0",
    "SeatCode": "2S1"
  },
  {
    "Column": 1,
    "Fare": "102",
    "NetFare": "102",
    "IsAvailableSeat": "True",
    "IsLadiesSeat": "False",
    "Length": 1,
    "Number": "4",
    "Row": 4,
    "Width": 1,
    "Zindex": 0,
    "Servicetax": "0",
    "OperatorServiceCharge": "0",
    "SeatCode": "3S1"
  },
  {
    "Column": 1,
    "Fare": "102",
    "NetFare": "102",
    "IsAvailableSeat": "True",
    "IsLadiesSeat": "False",
    "Length": 1,
    "Number": "5",
    "Row": 2,
    "Width": 1,
    "Zindex": 0,
    "Servicetax": "0",
    "OperatorServiceCharge": "0",
    "SeatCode": "4S1"
  },
  {
    "Column": 1,
    "Fare": "102",
    "NetFare": "102",
    "IsAvailableSeat": "True",
    "IsLadiesSeat": "False",
    "Length": 1,
    "Number": "6",

```



```

    "Row": 1,
    "Width": 1,
    "Zindex": 0,
    "Servicetax": "0",
    "OperatorServiceCharge": "0",
    "SeatCode": "5S1"
  }
],
"SeatsScript": null,
"BoardingTimes": null,
"DroppingTimes": null,
"Provider": "DZ/GmxpuN0rTfFNIdYpjWA==",
"ResponseStatus": 200,
"Message": null
}

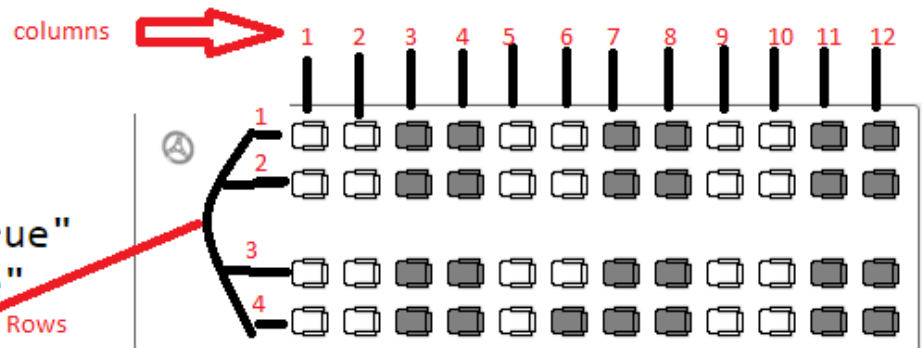
```

Below images describes to build seat layout:




```

{"TripId":"1046"
"Seats":[{"Column":1
"Fare":"2500"
"NetFare":"2500"
"IsAvailableSeat":"True"
"IsLadiesSeat":"False"
"Length":2
"Number":"U1"
"Row":1
"Width":1
"Zindex":0
"Servicetax":"375"
"OperatorServiceCharge":"10.00"}

```



For each seat, you will get these properties like length and width

- length = 1, width = 1  Seater
- length = 2, width = 1  Horizontal sleeper
- length = 1, width = 2  Vertical sleeper

Here Z-index indicates berth type:

- 0 - means lower berth
- 1 - means upper berth

5 Seat Blocking - BlockBusTicket()

5.1 Request URL

This API method is used to block the seat.

Method	URL
POST	Buses/BlockBusTicket

<http://webapi.i2space.co.in/Buses/BlockBusTicket>

5.2 Request Parameters

Parameter	Type	Required	Description
Address	string	REQUIRED	Address of passenger. Hyderabad
Ages	string	REQUIRED	Age of passenger. For single seat 24. Multiple seat 24~22.
BoardingId	string	REQUIRED	Id of Boarding place you selected. It is available in BoardingTimes object in AvailableBuses() or TripDetails(). 25000
BoardingPointDetails	string	REQUIRED	BoardingDetails of particular bus that you get in AvailableBuses() or TripDetails(). Chennai central
BusTypeName	string	REQUIRED	Refer BusTypes table. NON/AC , SEATER
CancellationPolicy	string	REQUIRED	Cancellation Policy of particular bus. 0:-0:100;6:-0:40;7:-0:30;8:-0:20;9:-0:0;10:-0:0
City	string	REQUIRED	City of passenger. Hyderabad
ConvenienceFee	string	REQUIRED	It is always 0. 0
DepartureTime	string	REQUIRED	DepartureTime of particular bus. 09:30 PM
DestinationId	string	REQUIRED	Destination ID of particular bus. 109
DestinationName	string	REQUIRED	Destination Name of particular bus. Bangalore
DisplayName	string	REQUIRED	Display Name of particular bus. GDS Demo Test
DroppingId	string	REQUIRED	Mandatory for RTCs. 12000
DroppingPointDetails	string	REQUIRED	Mandatory for RTCs. Bangalore central
EmailId	string	REQUIRED	Email ID of passenger. guru.m@i2space.com
EmergencyMobileNo	string	OPTIONAL	Emergency Mobile No of passenger. 9999999999
Fares	string	REQUIRED	Fare of particular seat that you selected from tripdetails method. For single seat 102. Multiple seat 102~102.
Genders	string	REQUIRED	Genders of passenger. M for Male, F for female. For single seat M. Multiple seat M~F.

IdCardNo	string	REQUIRED	IdCardNo of passenger. 142341789
IdCardType	string	REQUIRED	IdCardType of passenger. PAN_CARD
IdCardIssuedBy	string	REQUIRED	IdCardIssuedBy of passenger. Gov
JourneyDate	string	REQUIRED	Journey Date of particular bus. 11-11-2017
MobileNo	string	REQUIRED	Mobile No of passenger. 9999999999
Names	string	REQUIRED	Name of passenger. For single seat Guru. Multiple seat Guru~Gita.
NoofSeats	string	REQUIRED	Total no.of seats that you want to block. 1
Operator	string	REQUIRED	Travels of particular bus. GDS Demo Test
PartialCancellationAllowed	bool	REQUIRED	PartialCancellationAllowed value of particular bus. true
PostalCode	string	REQUIRED	PostalCode of passenger. 500035
Provider	string	REQUIRED	Provider of the particular bus. DZ/GmxpuN0rTfFNIdYpjWA==
ReturnDate	string	CONDITIONAL	Mandatory for roundtrip only. 12-11-2017
State	string	REQUIRED	State of passenger. Telangana
Seatcodes	string	CONDITIONAL	Seat code of particular seat. Mandatory if it is not null. For single seat 1S1. Multiple seat 2S1~3S1.
SeatNos	string	REQUIRED	Number of particular seat that you selected from TripDetails(). For single seat 3. Multiple seats split by tilt (~) like 3~4
Servicetax	string	REQUIRED	Service tax of particular seat that you selected from TripDetails(). For single seat 0. Multiple seat 0~0.
ServiceCharge	string	REQUIRED	Service charge of particular seat that you selected from TripDetails(). For single seat 0. Multiple seat 0~0.
SourceId	string	REQUIRED	Source Id of particular bus. 103
SourceName	string	REQUIRED	SourceName of particular bus. Chennai
Titles	string	REQUIRED	Title of passenger. For single seat Mr. Multiple seat Mr~Ms. Mr - Mister, Ms – Miss.
TripId	string	REQUIRED	Id parameter value that you get for every bus it is unique. 85
UserType	integer	REQUIRED	User type. Always 5 (STATIC). In XML request pass as Guest.

5.3 Sample Request

```
{
  "Address": "Hyderbad",
  "Ages": "24",
  "BoardingId": "25000",
  "BoardingPointDetails": "Chennai central",
  "BusTypeName": "NON/AC , SEATER",
  "CancellationPolicy": "0:-0:100;6:-0:40;7:-0:30;8:-0:20;9:-0:0;10:-0:0",
}
```

```

"City": "Hyderbad",
"ConvenienceFee": "0",
"DepartureTime": "09:30 PM",
"DestinationId": 109,
"DestinationName": "Bangalore",
"DisplayName": "GDS Demo Test",
"DroppingId": "12000",
"DroppingPointDetails": "Bangalore central",
"EmailId": "guru.m@i2space.com",
"EmergencyMobileNo": null,
"Fares": "102",
"Genders": "M",
"IdCardIssuedBy": "Gov",
"IdCardNo": "142341789",
"IdCardType": "PAN_CARD",
"JourneyDate": "11-11-2017",
"MobileNo": "9999999999",
"Names": "Guru",
"NoofSeats": "1",
"Operator": "GDS Demo Test",
"PartialCancellationAllowed": false,
"PostalCode": "500035",
"Provider": "DZ/GmxpuN0rTfFNIdYpjWA==",
"ReturnDate": null,
"Seatcodes": "1S1",
"SeatNos": "5",
"ServiceCharge": "0",
"Servicetax": "0",
"SourceId": 103,
"SourceName": "Chennai",
"State": "Telangana",
"Titles": "Mr",
"TripId": "85",
"TripType": 1,
"UserType": 5
}

```

5.4 Response Parameters

Parameter	Type	Description
BlockingReferenceNo	string	BlockingReferenceNo is reference number from operator end. 21086547
BookingReferenceNo	string	Reference Number from API. Required to book ticket. 1006902000
BookingStatus	integer	Defines status of ticket. Refer BookingStatus table. 1
Message	string	Defines whether blocking is success or failed. SUCCESS
ResponseStatus	integer	Refer at starting of the document. 200
previousFare	string	If IsPriceChange is false, you'll get null value. 589 For multiple seats: "589~589"
updatedFare	string	If IsPriceChange is false, you'll get null value. 618 For multiple seats: "618~618"

5.5 Sample Response

A) IsPriceChange: false

```
{
  "BlockingReferenceNo": "21086547",
  "BookingReferenceNo": "1006902000",
  "Message": "Blocked Seat Successfully",
  "IsPriceChange": false,
  "previousFare": null,
  "updatedFare": null,
  "BookingStatus": 1,
  "ResponseStatus": 200
}
```

B) IsPriceChange: true [As here price changes, you need to show latest price details to user before proceeding payment]

Below example is for UPSRTC, **Total Fare = BASIC_FARE + SERVICE_FEE + bookingFee**

```
{
  "BlockingReferenceNo": "As77S77knW",
  "BookingReferenceNo": "1007354688",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "589~589",
  "updatedFare": "618~618",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 100,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "618",
          "Type": "exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNB8tkP5I7ny7BCoo08GiJ9R8qXTd/qEnaUPe2n5rVy6ow=="
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ],
      "seatName": "16"
    },
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "618",
          "Type": "exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNB8tkP5I7ny7BCoo08GiJ9R8qXTd/qEnaUPe2n5rVy6ow=="
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ]
    }
  ]
}
```

```
    ],  
    "seatName": "15"  
  }  
],  
"TollFee": 0,  
"ReservationFee": 0  
}
```

NOTE For complete list of all RTCs implementation click on [RTC's Implementation](#)

6 Book Seat(s) - BookBusTicket()

This method is used to Book the ticket.

6.1 Request URL

Method	URL
GET	Buses/BookBusTicket?referenceNo=<referenceNo>

<http://webapi.i2space.co.in/Buses/BookBusTicket?referenceNo=1006902000>

6.2 Request Parameters

Parameter	Type	Required	Description
referenceNo	string	REQUIRED	Reference number of the particular ticket provided in block method reference. 1003506214

6.3 Response Parameters

Parameter	Type	Description
APIReferenceNo	string	PNR number for particular ticket. 110196433,1442222-5017184192
BookingStatus	integer	Status of the ticket. Refer BookingStatus table. 3
Message	String	Message. SUCCESS
OperatorPNR	string	For future purpose. 5017184192
ReferenceNo	string	Reference number from API. 1006902000

6.4 Sample Response

```
{
  "OperatorPNR": "5017184192",
  "ReferenceNo": "1006902000",
  "APIReferenceNo": "110196433,1442222-5017184192",
  "Message": "SUCCESS",
  "BookingStatus": 3,
  "ResponseStatus": 200,
  "RefundResponse": null,
  "Provider": null
}
```

7 Cancellation Details - CancellationDetails()

This method is used for getting cancellation details.

7.1 Request URL

Method	URL
GET	Buses/CancellationDetails?referenceNo=<referenceNo>&seatNos=<seatNos>&emailId=<emailId>

<http://webapi.i2space.co.in/Buses/CancellationDetails?referenceNo=1006902000&seatNos=5&emailId=guru.m@i2space.com>

7.2 Request Parameters

Parameter	Type	Required	Description
referenceNo	string	REQUIRED	Reference number of the particular ticket provided in book method reference. 1006902000
seatNos	string	REQUIRED	Seat no of the particular reference number. 5
emailId	string	REQUIRED	Email id provided during the blocking method for this particular ticket. guru.m@i2space.com

7.3 Response Parameters

Parameter	Type	Description
BookingStatus	integer	Status of the ticket. Refer BookingStatus table. 5
TicketNumber	string	Ticket number. 111040378-1495231
TotalTicketFare	string	Total ticket fare. 102
TotalRefundAmount	string	Total refund amount. 102
Message	string	Message. Cancellation is possible.
CancellationCharges	string	Cancellation charges. 0
RefundType	string	Refund type. null
isSeatCancellable	boolean	Is seat cancellable. true
PartialCancellationAllowed	boolean	Partial cancellation allowed. false
RefundResponse	string	Refund response. null

7.4 Sample Response

```
{
  "BookingStatus": 5,
  "TicketNumber": "111040378-1495231",
  "TotalTicketFare": 102,
  "TotalRefundAmount": 102,
```



```
"Message": "Cancellation is possible.",  
"CancellationCharges": 0,  
"RefundType": null,  
"isSeatCancellable": true,  
"PartialCancellationAllowed": false,  
"RefundResponse": null  
}
```

8 Cancel Bus Ticket - CancelBusTicket()

This method is used for cancelling bus ticket.

8.1 Request URL

Method	URL
GET	Buses/CancelBusTicket?referenceNo=<referenceNo>&seatNos=<seatNos>&emailId=<emailId>

<http://webapi.i2space.co.in/Buses/CancelBusTicket?referenceNo=1006902000&seatNos=5&emailId=guru.m@i2space.com>

8.2 Request Parameters

Parameter	Type	Required	Description
referenceNo	string	REQUIRED	Reference number of the particular ticket provided in book method reference. 1003506214
seatNos	string	REQUIRED	Seat no of the particular reference number. 1
emailId	string	REQUIRED	Email id provided during the blocking method for this particular ticket. test@gmail.com

8.3 Response Parameters

Parameter	Type	Description
BookingStatus	integer	Status of the ticket. Refer BookingStatus table. 5
CancellationCharges	double	Cancellation charges. 240.0
Message	string	Message. Ticket Cancelled Successfully for seats 1
RefundStatus	integer	Status of cancelled ticket amount refund. 2
TicketNumber	string	PNR number of the ticket. I2S1356
TotalRefundAmount	double	Total refund amount after cancellation. 960
TotalTicketFare	double	Total amount of the ticket. 1200

8.4 Sample Response

```
{
  "BookingStatus": 5,
  "TicketNumber": "110196433-1442222",
  "TotalTicketFare": 102,
  "TotalRefundAmount": 102,
  "Message": "Ticket cancelled successfully for seats 5.",
  "CancellationCharges": 0,
```

```
"RefundType": null,  
"isSeatCancellable": false,  
"PartialCancellationAllowed": false,  
"RefundResponse": {  
  "TransactionId": null,  
  "PaymentId": null,  
  "Message": null,  
  "Amount": 0,  
  "Mode": null,  
  "PgRefNo": null,  
  "RefundStatus": 2,  
  "RequestDatatime": "0001-01-01T00:00:00"  
}  
}
```

9 BusTicketBookingDetails()

9.1 Request URL

This method is used to check the Ticket Details.

Method	URL
GET	Buses/BusTicketbookingDetails?referenceNo=<referenceNo>&type=<type>

<http://webapi.i2space.co.in/Buses/BusTicketbookingDetails?referenceNo=1006902000&type=2>

9.2 Request Parameters

Parameter	Type	Required	Description
referenceNo	string	REQUIRED	Reference Number. 1003506214
type	string	REQUIRED	A static value which is used to get ticket details. 2

9.3 Sample Response

```
{
  "PSAName": "",
  "ActualFare": "102.00",
  "CollectedFare": "102.00",
  "BookingDate": "07 Nov 2017, 11:59 AM",
  "BookingStatus": "Booked",
  "SMSUsageCount": 0,
  "CancellationRemarks": "-",
  "PassengerName": "Guru",
  "APIRefNo": "110196433,1442222-5017184192",
  "ClientID": 1328,
  "CreatedById": 3490,
  "EcommerceSegment": [ ],
  "PaymentInfo": {
    "PaymentName": null,
    "PaymentId": null,
    "ResponseCode": null,
    "ResponseMessage": null,
    "MerchtRefNumber": null,
    "Amount": null,
    "Mode": null,
    "TransactionId": null,
    "PaymentMethod": null,
    "PaymentStatus": 1
  },
  "PartnerAgentDetails": {
    "AgentId": "",
    "ServiceId": null,
    "ServiceName": null,
    "Amount": null,
    "UserCharges": null,
    "ResCode": null,
    "ResDesc": null,
    "Requestid": ""
  }
},
```

```

"DeviceModel": null,
"DeviceOS": null,
"DeviceOSVersion": null,
"DeviceToken": null,
"ApplicationType": 0,
"ConvenienceFee": "0",
"IsAgentPaymentGateway": false,
"City": null,
"State": null,
"IsOfflineBooking": false,
"TripId": "85",
"BoardingId": "25000",
"DroppingId": null,
"NoofSeats": "1",
"Fares": "102",
"Servicetax": "0",
"ServiceCharge": "0",
"SeatNos": "5",
"Titles": "Mr",
"Names": "Guru",
"Ages": "24",
"Genders": "M",
"Address": "Hyderabad",
"PostalCode": "500035",
"BookingRefNo": "1006902000",
"IdCardType": null,
"IdCardNo": null,
"IdCardIssuedBy": null,
"MobileNo": "9999999999",
"EmergencyMobileNo": null,
"EmailId": "guru.m@i2space.com",
"Provider": "DZ/GmxpuN0rTfFNidYpjWA==",
"Operator": "GDS Demo Test",
"BlockId": "21086547",
"PartialCancellationAllowed": false,
"BoardingPointDetails": "Chennai central-09:30 PM",
"BoardingDroppingDetails": {
  "Address": "",
  "ContactNumbers": "",
  "ContactPersons": "",
  "PointId": null,
  "Landmark": "",
  "Location": "Chennai central",
  "Name": "",
  "Time": ""
},
"DroppingPointDetails": null,
"DroppingDetails": {
  "Address": "",
  "ContactNumbers": "",
  "ContactPersons": "",
  "PointId": null,
  "Landmark": "",
  "Location": "",
  "Name": "",
  "Time": ""
},
"BusTypeName": "NON/AC , SEATER",
"DepartureTime": "09:30 PM",
"PaymentId": "",
"CancellationTime": null,
"CancellationPolicy": "0:-0:100;6:-0:40;7:-0:30;8:-0:20;9:-0:0;10:-0:0",

```

```
"PromoCode": null,  
"PromoCodeAmount": 0,  
"PostMarkup": 0,  
"EProductPrice": 0,  
"Ecommerceproducts": [ ],  
"DisplayName": "GDS Demo Test",  
"Seatcodes": null,  
"PreviousFare": null,  
"IsWallet": false,  
"CurrencyID": "INR",  
"CurrencyValue": "1",  
"Remarks": "",  
"IsPartnerAgentDetails": null,  
"SourceId": "103",  
"SourceName": "Chennai",  
"DestinationId": "109",  
"DestinationName": "Bangalore",  
"JourneyDate": "11-11-2017",  
"ReturnDate": "",  
"TripType": 1,  
"BusType": null,  
"OperatorId": null,  
"OperatorName": "GDS Demo Test",  
"BusTimings": null,  
"NoOfSeatsRequired": null,  
"User": "YollTxsxtJwEWRJIqKAvsA==",  
"UserType": 5,  
"AffiliateId": null,  
"WebsiteUrl": null  
}
```

10 BoardingPointDetails()

10.1 Request for BoardingPointDetails

This API method provides Available Buses in particular route for particular date.

Method	URL
GET	<code>http://webapi.i2space.co.in/Buses/BoardingPointDetails?boardingPoint=<boardingPoint>&provider=<provider>&tripid=<tripid></code>

`http://webapi.i2space.co.in/Buses/BoardingPointDetails?boardingPoint=2238&provider=ASw7iG9082iyI7qZCJasgA==&tripid=19369-0`

10.2 Request Parameters

Parameter	Type	Required	Description
boardingPoint	string	REQUIRED	Boarding Point of particular bus. 2238
provider	string	REQUIRED	Provider of particular bus. ASw7iG9082iyI7qZCJasgA==
tripid	string	REQUIRED	Id of the particular bus. 19369-0

10.3 Response

```
{
  "Address": "Ameerpet, Jeans Corner",
  "ContactNumbers": "9999999999",
  "Landmark": "",
  "Location": "Ameerpet, Jeans Corner",
  "Time": null
}
```

NOTE For few operators you may not get the BoardingPointDetails. In such case you can consider the boarding points which you get in availability/tripdetails method.

11 Check API Balance - GetAPIBalance()

11.1 Request

This method is used to get the Account Balance Status.

Method	URL
GET	Accounts/GetApiBalance

<http://webapi.i2space.co.in/Accounts/GetApiBalance>

11.2 Response

```
{  
  Balance: "24641.26",  
  Message: "Sucess",  
  StatusCode: 200  
}
```


12 RTCs Implementation

12.1 Overview

This method is used to implement various RTC operators listed below:

Operator Names:

- PEPSU
- OSRTC
- JKSRTC
- WBSTC
- BSTDC
- KTCL
- HRTC
- UPSRTC
- GSRTC

Kindly make below changes at your end in order to consume RTCs:

- a. Both boarding point and dropping point is mandatory for RTC bookings.
- b. Fare will change basis boarding point and dropping point selected for an OSRTC and JKSRTC service.
- c. We recommend you to use the attached ticket template for hard copy of OSRTC and JKSRTC tickets.
- d. PEPSU(PRTC), OSRTC and JKSRTC might not honor the booking if the customer produces the ticket print out which is not the same as the one shared with you.

Need to implement below changes in API:

Please follow below steps to consume RTC inventory at your end:

1. Call **AvailableBuses()** as usual. You will see the PEPSU(PRTC) or PEPSUPRTC, OSRTC and JKSRTC services (if available) with pick up and drop points. Note that the fare that you get on invoking this method is not the actual fare.
2. No changes in **TripDetails()** method.
3. Invoke the **BlockBusTicket()** method with dropping point in addition to other details (like name, contact, boarding point etc.) as input parameter. The parameters need to pass are **DroppingPointDetails**, **DroppingId**.
4. In Blocking response, you will get the updated fare, you need to compare with the parameter **IsPriceChange** (If it is true then you will need to fetch the updated fare else the old fare). This will give you the actual fare of the service for the selected boarding point and dropping point in case of PEPSU(PRTC), OSRTC and JKSRTC services.
5. Now invoke the **BookBusTicket()** method with the same Block Reference number as the input parameter.
6. You need to calculate the total fare within FareBreakUps section for RTCs.

Points at Glance:

- Both Boarding point and Dropping point are mandatory for all RTCs.
- You can get the Dropping Point details in same methods where you are getting the Boarding Point Details in **AvailableBuses()** and **TripDetails()** methods.
- You will get Boarding point details in **BoardingTimes** Array and Dropping point details in **DroppingTimes** Array.
- In below sections you can find blocking response and the total fare that needs to show to user.

NOTE Dropping Point Details string should be in a format as "DroppingDetails - DroppingTime"

12.2 UPSRTC

Total Fare = BASIC_FARE + SERVICE_FEE + bookingFee

```
{
  "BlockingReferenceNo": "As77S77knW",
  "BookingReferenceNo": "1007354688",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "589~589",
  "updatedFare": "618~618",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 100,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "618",
          "Type":
"exDjudygzG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNB8tkP5I7ny7BCoo08GiJ9R8qXTd/qEnaUPe2n5rVy6ow=="
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ],
      "seatName": "16"
    },
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "618",
          "Type":
"exDjudygzG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNB8tkP5I7ny7BCoo08GiJ9R8qXTd/qEnaUPe2n5rVy6ow=="
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ],
      "seatName": "15"
    }
  ],
  "TollFee": 0,
  "ReservationFee": 0
}
```

12.3 HRTC

Total Fare = BASIC_FARE + LEVIES_CHARGES + TOLL_FEE + SERVICE_FEE + RESERVATION_FEE + CONCESSIONS_FEE

```
{
  "BlockingReferenceNo": "ZvbROtI6jM",
  "BookingReferenceNo": "1003354695",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "898~898",
  "updatedFare": "898~898",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 0,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "898",
          "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNCpCqjnZDfwLuyZ9dyA3hL52NOjz oj+1WVcAH2GrvCs9g==
"
        },
        {
          "componentName": "LEVIES_CHARGES",
          "value": "11.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        },
        {
          "componentName": "TOLL_FEE",
          "value": "17.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8Sq8pD9mjDbGD+JGyqxYkxZB9zMaeCxrsr9ST92TWd+s"
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "14.0",
          "Type": "glhnZEi4jrdpWh0+1+Vja+NQsWStz3zDXHGWf24+eAhEAm3veBdC4QDBxeCdIqgz"
        },
        {
          "componentName": "RESERVATION_FEE",
          "value": "0.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        },
        {
          "componentName": "CONCESSIONS_FEE",
          "value": "0.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ],
      "seatName": "14"
    },
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "898",
          "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNCpCqjnZDfwLuyZ9dyA3hL52NOjz oj+1WVcAH2GrvCs9g==
"
        },
        {
          "componentName": "LEVIES_CHARGES",
          "value": "11.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ],
    }
  ]
}
```

```
{
  {
    "componentName": "TOLL_FEE",
    "value": "17.0",
    "Type": "FyFfYnCaFp+RM/S+uY7p8Sq8pD9mjDbGD+JGyqxYkxZB9zMaeCxrsr9ST92TWd+s"
  },
  {
    "componentName": "SERVICE_FEE",
    "value": "14.0",
    "Type": "glhnZEi4jrdpWh0+1+Vja+NQsWStz3zDXHGwf24+eAhEAm3veBdC4QDBxeCdIqgz"
  },
  {
    "componentName": "RESERVATION_FEE",
    "value": "0.0",
    "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
  },
  {
    "componentName": "CONCESSIONS_FEE",
    "value": "0.0",
    "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
  }
],
"seatName": "13"
}
],
"TollFee": 0,
"ReservationFee": 0
}
```

12.4 JKSRTC

Total Fare: BASIC_FARE + ServiceCharge + LEVIES_CHARGES

```
{
  "BlockingReferenceNo": "a70f70P5eS",
  "BookingReferenceNo": "1004454698",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "86~86",
  "updatedFare": "86~86",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 0,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "86",
          "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNvruZKde9jA9t5WFeUZQg2nGMv/ihNLtTy7yGOYn+Bcw=="
        },
        {
          "componentName": "ServiceCharge",
          "value": "3.7",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ZE8w2dl2BeYqPTzwnSL5OE+iNPJm7vE2/+YJpEU73je"
        },
        {
          "componentName": "LEVIES_CHARGES",
          "value": "6.5",
          "Type": "FyFfYnCaFp+RM/S+uY7p8c2fdFM2mVCkxLJsU32DNH/U2/tcviFLvUlmrLd+C6dz"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "seatName": "13"
},
{
  "CustomerPriceBreakUp": [
    {
      "componentName": "BASIC_FARE",
      "value": "86",
      "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNAVruZKde9jA9t5WFeUZQg2nGMv/ihNLtTy7yGOYn+Bcw=="
    },
    {
      "componentName": "ServiceCharge",
      "value": "3.7",
      "Type": "FyFfYnCaFp+RM/S+uY7p8ZE8w2dl2BeYqPTzwnSL5OE+iNPJm7vE2/+YJpEU73je"
    },
    {
      "componentName": "LEVIES_CHARGES",
      "value": "6.5",
      "Type": "FyFfYnCaFp+RM/S+uY7p8c2fdFM2mVCkxLJsU32DNH/U2/tcviFLvUlmrLd+C6dz"
    }
  ],
  "seatName": "14"
}
],
"TollFee": 0,
"ReservationFee": 0
}

```

12.5 OSRTC

Total Fare = BASIC_FARE + ServiceCharge

```

{
  "BlockingReferenceNo": "RaJHmcD2oY",
  "BookingReferenceNo": "1005954697",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "393~393",
  "updatedFare": "412~412",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 0,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "412",
          "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNDfIIiJlqAJot1bMAkXN4iHS2jrQ5+/YXWRdM0x0M+2vw=="
        },
        {
          "componentName": "ServiceCharge",
          "value": "20.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
        }
      ]
    }
  ],

```

```

    "seatName": "14"
  },
  {
    "CustomerPriceBreakUp": [
      {
        "componentName": "BASIC_FARE",
        "value": "412",
        "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNDfIIiJlqAJot1bMAkXN4iHS2jrQ5+/YXWRdM0x0M+2vw=="
      },
      {
        "componentName": "ServiceCharge",
        "value": "20.0",
        "Type": "FyFfYnCaFp+RM/S+uY7p8ePzFy6xsUtNXsungf2uzrYeNiRX3a/VtvrSnkOsRh1H"
      }
    ],
    "seatName": "13"
  }
],
"TollFee": 0,
"ReservationFee": 0
}

```

12.6 PEPSU

Total Fare = BASIC_FARE + SERVICE_FEE + OTHER_CHARGES

```

{
  "BlockingReferenceNo": "dsCbZdMFcq",
  "BookingReferenceNo": "1006954693",
  "Message": "SUCCESS",
  "IsPriceChange": true,
  "previousFare": "575~575",
  "updatedFare": "575~575",
  "BookingStatus": 1,
  "ResponseStatus": 200,
  "bookingFee": 0,
  "FareBreakUps": [
    {
      "CustomerPriceBreakUp": [
        {
          "componentName": "BASIC_FARE",
          "value": "575",
          "Type":
"exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNBMbPT6C+u5teqrukMttqc4e0KfvaM89bwhGF2wojd/IJA=="
        },
        {
          "componentName": "SERVICE_FEE",
          "value": "14.75",
          "Type": "glhnZEi4jrdpWh0+1+Vja+NQsWStz3zDXHGwf24+eAhEAm3veBdC4QDBxeCdIggz"
        },
        {
          "componentName": "OTHER_CHARGES",
          "value": "15.0",
          "Type": "FyFfYnCaFp+RM/S+uY7p8Sg8pD9mjDbGD+JGyqxYkxYWzaxy1W3rT+reCZKQsK11"
        }
      ],
      "seatName": "16"
    }
  ],
}

```

```
{
  "CustomerPriceBreakUp": [
    {
      "componentName": "BASIC_FARE",
      "value": "575",
      "Type": "exDjudygZG2yCubSMKqX+KdgjS2KEdhN6HvPitcGWNBMBpT6C+u5tegrukMtc4e0KfvaM89bwhGF2wojd/IJA=="
    },
    {
      "componentName": "SERVICE_FEE",
      "value": "14.75",
      "Type": "g1hnZEi4jrdpWh0+1+Vja+NQsWStz3zDXHGwf24+eAhEAm3veBdC4QDBxeCdIggz"
    },
    {
      "componentName": "OTHER_CHARGES",
      "value": "15.0",
      "Type": "FyFfYnCaFp+RM/S+uY7p8Sq8pD9mjDbGD+JGyqxYkxYWzaxy1W3rT+reCZKQsK11"
    }
  ],
  "seatName": "15"
},
"TollFee": 0,
"ReservationFee": 0
}
```

13 Certification & Guidelines

13.1 Certification

Client needs to go through certification process in order to get LIVE credentials by providing reference numbers for below test cases.

[Case 1]: One-way single seat booking.

[Case 2]: One-way multiple seats booking.

[Case 3]: Round-trip single seat booking.

[Case 4]: Round-trip multiple seats booking.

[Case 5]: Cancellation reference numbers.

13.2 [Case 1]: One-way single seat booking

Please follow below algorithm for one-way trip:

- a. Get cities list from Sources method
- b. Choose one bus from AvailableBuses method
- c. Get seat details from TripDetails method (from the response b)
- d. Go through BlockBusTicket method (from the responses b, c)
- e. If payment is successful, go through BookBusTicket method (from the response d)

13.3 [Case 2]: One-way multiple seats booking

In this case, everything is same as one-way single seat booking. But the difference is there in BlockBusTicket method that you need to pass few below mentioned parameters by separating with tilt (~) symbol.

No of seats parameter should contain total no of seat you want book. If anyone parameter went wrong you can't book ticket.

```
{
  "Ages": "24~22",
  "Fares": "102~102",
  "Genders": "M~F",
  "Names": "Guru~Gita",
  "NoofSeats": "2",
  "Seatcodes": "2S1~3S1",
  "SeatNos": "3~4",
  "ServiceCharge": "0~0",
  "Servicetax": "0~0",
  "Titles": "Mr~Ms"
}
```

13.4 [Case 3]: Round-trip single seat booking

It is similar one-way trip but need to hit every method for 2 times. Please go through below algorithm for roundtrip.

- a. Get cities list from Sources method
- b. Choose onward bus from AvailableBuses method.
- c. Get seat details of onward bus from TripDetails method (from the response b, show to user)
- d. Choose return bus from AvailableBuses method.
- e. Get seat details of return bus from TripDetails method (from the response d, show to user)

- f. Go through BlockBusTicket method for onward bus (from the responses b, c)
- g. Go through BlockBusTicket method for return bus (from the responses d, e)
- h. If payment is successful, go through BookBusTicket method for onward bus (from the response f)
- i. If payment is successful, go through BookBusTicket method for return bus (from the response g)

13.5 [Case 4]: Round-trip multiple seats booking

It is same similar as Round-trip (need to hit every method for 2 times individually as mentioned in Case 3) but need to pass multiple values with tilt(~) separation(as mentioned in Case2).

13.6 [Case 5]: Cancellation reference numbers

In CancelBusTicket method, already gone through single seat cancellation. While you are cancelling multiple seats send the seat number with Comma (,) (**REQUIRED**) separate as below URL.

```
http://webapi.i2space.co.in/Buses/CancelBusTicket?referenceNo=1007802002&seatNos=3,4&emailId=guru.m@i2space.com
```

14 FAQ(s)

14.1 MISC

Q. What is flow of One-way and Round-Trip Booking?

You need to follow below guidelines:

One-way-Trip:

- a. Get cities list from Sources method
- b. Choose one bus from AvailableBuses method
- c. Get seat details from TripDetails method (from the response b)
- d. Go through BlockBusTicket method (from the responses b, c)
- e. If payment is successful, go through BookBusTicket method (from the response d)

Round-Trip:

- a. Get cities list from Sources method
- b. Choose onward bus from AvailableBuses method.
- c. Get seat details of onward bus from TripDetails method (from the response b, show to user)
- d. Choose return bus from AvailableBuses method.
- e. Get seat details of return bus from TripDetails method (from the response d, show to user)
- f. Go through BlockBusTicket method for onward bus (from the responses b, c)
- g. Go through BlockBusTicket method for return bus (from the responses d, e)
- h. If payment is successful, go through BookBusTicket method for onward bus (from the response f)
- i. If payment is successful, go through BookBusTicket method for return bus (from the response g)

Q. How to call your web service?

You need to create and send a HTTP POST/GET request for the API. For more details please refer to our API specification documents.

Q. Why the URLs mentioned in the API document are not working?

URL mentioned in the document should pass along with header key as mentioned. It can't be done from browser itself. You can use api-testing tool/add-on like post-man (recommended), rest-client etc.... & also response will throw "HTTP Error 400. The request is badly formed", because there might be chances of spaces in the URL.

Q. We want to add amenities to our services, how to get that at our website?

Currently, we're not getting amenities from our suppliers. When we get update on amenities we'll notify you.

Q. How to resolve error message "400: Bad Request"?

This error message comes in 2 cases:

- 1) RequestURL is not valid (Please check if any unnecessary white-spaces are there)
- 2) RequestBody is not valid (Please validate your JSON request at <http://jsonlint.com>)

Q. What is the procedure to get the live credentials of your web service?

First of all, you need complete all test cases. Business team will revert with these details while in commercial agreement phase.

Q. After we get LIVE credentials what needs to change at our end?

Business Team will provide you ConsumerKey, ConsumerSecret and LIVE URL (instead of <http://webapi.i2space.co.in>).

Q. We have changed our server, do we need to make any changes in API?

Kindly make sure that your server is PUBLIC and you are using your own credentials. Your server IP must be white-listed at our end.

Q. What will be terms and conditions we should put before a passenger can book a ticket?

You'll be able to book only 6 seats max in one transaction. However, this is not applicable for RTC's especially (E.g. For Uttarakhand RTC, it accepts only one seat per transaction).

Q. Risk and compliance if any please specify?

For API, please do not share your consumer key and consumer secret without any intervention from our side, we do not have any risk/compliance for API partners as of now. We'll let you know, if there are any.

14.2 Sources()

Q. Do I need to hit your API every time for sources?

No. We're not recommending you to hit our API for every time. If you hit API every time for sources, burden on API server will increase which leads to response delay.

At first, you can save all sources in information in your local file. You can access sources list from that file. Later you can update sources list periodically for every 1 or 2 months.

14.3 AvailableBuses()

Q. Do all routes give results of Available Buses?

No. We can't get available buses from all routes. We're recommending few routes for available buses:

- Hyderabad to Bangalore & Bangalore to Hyderabad
- Chennai to Bangalore & Bangalore to Chennai

In LIVE, you'll get available buses for all routes.

Q. Unable to get the bus available for particular route or on particular dates?

If you are unable to get the Buses Availability, means there are no buses available for that particular route or on particular date

Suggestion: If no buses available, please try for another date/route.

Q. How to translate the cancellation policy string?

The entire string is a comma separated list of having four parts separated by semi colon. "From Time: To Time:

Cancellation Rate: percentage or absolute." e.g. "0:12:100:0; 12:-1:10:0"

0:12:100:0; means if the customer cancels the ticket within 12 hours of departure time then cancellation amount will be 100%. (e.g. departure time: 8:00 pm and if we cancel anytime between 8:00am to 8:00 pm then cancellation amount will be 100%)

12:-1:10:0 means if the customer cancel the ticket any time before 12 hours of departure time then cancellation amount will be 10%. (e.g. departure time: 8:00 pm and if we cancel any time before 8:00 am then cancellation amount is 10%.) :0 represents Percentage...

Q. How to convert the below time formats to exact time format like 10:15 A.M, 11:25 P.M etc.

1.[arrivalTime] => 510 && 2.[departureTime] => 1140

Time Format are explained in minutes. We are providing time in Minutes for all the Buses in 'Available Trips' from the start of the day. Minutes should be divided by 60 to get the time formats. Consider the below two examples

The time starts from 00:00am.

<p>(i) for example, if the dropping time is 1095 then $1095/60 = 18$ and remainder is 15 Journey Day = $18/24 = 0$ Hour = $18\%24 = 18$, Minutes = 15 => 18 hrs 15 min => 06:15 pm (same day as doj)</p>	<p>(ii) for example, if the dropping time is 1600 then $1600/60=26$ and remainder is 40 Journey Day = $26/24 = 1$ Hour = $26\%24 = 2$, Minutes = 40 =>2:40am of the next day(next day from doj)</p>
--	---

Q. What is the significance of Mticket flag which we are sending in available trips?

The Mticket flag which is being sent in AvailableBuses() method has to be implemented at partner's end.

If this flag is TRUE for a particular operator then Mticket (SMS ticket) will be considered valid and mobile ticket can be produced at the time of journey for that particular operator.

Q. Why we are getting different fares in the AvailableBuses() method and TripDetails () method?

The fare coming in AvailableBuses() method is the generic fare for the service and in TripDetails() method you will get fare of individual seats , it may vary due to seat types.

Q. How to get one way and round-trip details?

A: There is a parameter called triptype which defines whether it is one way or round trip i.e., triptype=1(one way), triptype=2(roundtrip). You need to pass those parameter in availability request.

One way:

```
http://webapi.i2space.co.in/Buses/AvailableBuses?sourceId=103&destinationId=109&journeyDate=11-11-2017&tripType=1&userType=5&user=&returnDate=12-11-2017
```

RoundTrip:

```
http://webapi.i2space.co.in/Buses/AvailableBuses?sourceId=103&destinationId=109&journeyDate=11-11-2017&tripType=2&userType=5&user=&returnDate=12-11-2017
```

14.4 TripDetails()

Q. In Trip details methods if we get seatscript as null or empty?

It means at that particular time from operator end you may not get the seat layout. The seats value might be empty in response of API so that the seats are not displayed.

Suggestion: Maintain a popup box with message as "Oops something went wrong from operator end, Please select another operator".

Q. Why we are getting different fares in the AvailableBuses () method and TripDetails () method?

The fare coming in AvailableBuses() method is the generic fare for the service and in TripDetails() method you will get fare of individual seats, it may vary due to seat types.

Q. Is there any method to call seat layout?

The seat layout needs to be generated using the seat array map returned from the trip details.

To begin with there are three different kinds of seats:

Seater (length = 1, width =1)

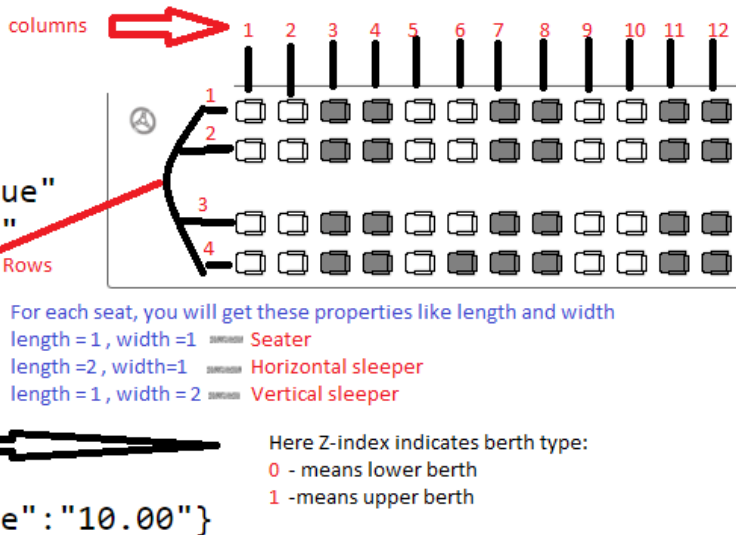
Horizontal sleeper (length =2, width=1)

And Vertical sleeper (length = 1, width = 2)

Here z indicates upper/lower berths. If z=0(lower berth) and z=1(upper berth).

```

{
  "TripId": "1046"
  "Seats": [
    {
      "Column": 1
      "Fare": "2500"
      "NetFare": "2500"
      "IsAvailableSeat": "True"
      "IsLadiesSeat": "False"
      "Length": 2
      "Number": "U1"
      "Row": 1
      "Width": 1
      "Zindex": 0
      "Servicetax": "375"
      "OperatorServiceCharge": "10.00"
    }
  ]
}
```



columns

Rows

For each seat, you will get these properties like length and width

- length = 1, width =1 — Seater
- length =2, width =1 — Horizontal sleeper
- length = 1, width = 2 — Vertical sleeper

Here Z-index indicates berth type:

- 0 - means lower berth
- 1 - means upper berth

Q. How to differentiate Seat Types?

Based on **BusType** parameter value, programmatically you can differentiate Seat Types as **AC Semi-Sleeper**, **AC Sleeper**, **Non-AC Semi-Sleeper** and **Non-AC Sleeper**.

14.5 BlockBusTicket ()

Q. How to confirm whether ticket is blocked or not?

When the ticket is blocked you will get a response which contains BookingReferenceNo.

```

{
  "BlockingReferenceNo": "21086547",
  "BookingReferenceNo": "1006902000",
  "Message": "Blocked Seat Successfully",
  "IsPriceChange": false,
  "previousFare": null,
  "updatedFare": null,
  "BookingStatus": 1,
  "ResponseStatus": 200
}
```

Q. Unable to block the ticket?

Blocking request is case sensitive and is in standard format. If you are unable to fill all the parameters or by entering any mismatch/wrong data the ticket will not be blocked.

Q. How can I block multi seats?

If you want to block the ticket for more than one passenger, you have to send the block request with few parameters with tilt separator (~).

```
{
  "Ages": "24~22",
  "Fares": "102~102",
  "Genders": "M~F",
  "Names": "Guru~Gita",
  "NoofSeats": "2",
  "Seatcodes": "2S1~3S1",
  "SeatNos": "3~4",
  "ServiceCharge": "0~0",
  "Servicetax": "0~0",
  "Titles": "Mr~Ms"
}
```

NOTE No of seats parameter should contain total no of seat you want book if any one parameter went wrong you can't book ticket.

Q. Status response for Blocking, Booking and Cancelling.

In response you will get bookingstatus based on status you need to proceed for next process and messages need to customized at your end if required.

BookingStatus=1 =>Blocked

BookingStatus=2=>BlockingFailed

BookingStatus=3=>Booked

BookingStatus=4=>BookingFailed

BookingStatus=5=>Cancelled

Suggestion:

if BookingStatus!=1

return with customized message or as "Please Contact Administrator" message.

14.6 BookBusTicket()

Q. How can we do test bookings? Can we have testing credentials?

We provide testing credentials if required, so that you can do all the test bookings by using provided testing credentials.

Q. How long we can delay in booking after blocking a seat?

You can confirm blocked seats in another 10 minutes of time span, otherwise you may face error as 'SEAT NO MORE AVAILABLE' error.

Q. We are facing number of "TENTATIVE BOOKING FAILED" errors, what are the causes and how can we reduce it?

CAUSES FOR TENTATIVE BOOKING FAILED:

There can be external dependencies like changes at operator's end, then 'TENTATIVE BOOKING FAILED ERROR' may occur.

- Whenever the seat selected is a 'LADIES SEAT', but the passenger's title is given as 'Mr' or the gender is given as 'MALE', then 'TENTATIVE BOOKING FAILED ERROR' occurs.
- Whenever the data which partners are accessing is stale, then 'TENTATIVE BOOKING FAILED ERROR' occurs.
- ID proof is mandatory in some cases, but idtype and idnumber is mandatory in all cases, so even if there is no value to be entered the fields must be entered in the JSON request.

HOW TO REDUCE TENTATIVE BOOKING FAILED ERROR...???

- Kindly check your JSON request, check if you have mentioned all the necessary fields and given required values.
- Make sure if you are booking a lady's seat, it should be booked with title 'Ms' or 'Mrs' and the gender should be mentioned as 'FEMALE'.
- Keep refreshing your database every 3-4 hours for all the updates at our end.
- Make sure that the IDtype must be written as PAN_CARD, VOTER_CARD, PASSPORT, DRIVING_LICENCE, RATION_CARD, and AADHAR.

14.7 CancelBusTicket()

Q. Can I cancel the primary passenger?

If partial cancellation is not allowed from operator's end you are not authorized to cancel the seat for a particular passenger.

.

APPENDIX - Error Codes

Code	Message	Description
200	Success	This message indicates Successful completion of a request
400	Not a valid request	One or more parameters for the request has been missed out
401	Authentication failed	API Key is blank or not valid
401/403	Access Denied / Unauthorized	If the request Content Type is NOT application/JSON. Unauthorized when consumer key and consumer key are not Matching
500	Internal Server Error	Errors, Exceptions if any occurs in Server Side while processing a request
500	ERROR:Fares mismatch.	Please check the fares values are sending properly in blocking request. When making multi-seat booking check all parameters that we recommended.



A Cloud & API Travel Technology Platform powered by i2Space Web Technologies Private Limited with a business presence in emerging markets such as South-East Asia, Middle East & Africa. With more than 10 services offered to 300+ clients with 1000s of Agents, eTravos is one of the fastest growing companies in the Cloud & API Travel Technology Platform. eTravos is built upon on the cutting-edge Cloud Computing Technologies where everything starting from the plug-and-play Web-APIs to Language, Currencies and Payment Gateways can be customized almost instantly.

Either it is a small or large business, choose eTravos to sell travel tickets. Because of its ease of website and mobile apps setup, remarkable features, and global industry-leading ability to grow your business easily.

“Everything you need Cloud Travel Services”

For any technical assistance, mail us at

TechnicalSupport@i2space.com