Singapore Airlines
AppChallenge
API Specification Document

Contents

k

1. Objective 3
2. Definitions and Abbrieviations 3
3. Overview on Flow and Connection Details 3
3.1 System Endpoints 3
3.2 Accessing APIs 3
3.3 Overall Flow 3
4 API List & Specifications 6
4.1 Flight Search 7
4.2 GetFare 9
4.3 Create PNR 11
4.4 Get PNR 12
4.4 Get Seat Map 14
4.5 Select Seat 17
4.6 Extend Session 18
4.6 Confirm PNR 19
4.7 Krisflyer Profile 21
4.8 Checkin GetPassenger 23
4.9 Checkin UpdatePassenger 26
4.10 Checkin Checkin 27
4.11 Checkin CancelCheckin 28
4.12 Checkin GetBoardingPass 29

1. Objective

This document defines the detailed specifications for the SIA APIs being exposed for the purpose of the AppChallenge.

2. Definitions and Abbrieviations

Acronym	Description
PNR	Passenger Name Record
RBD	Reservation Booking Designator

3. Overview on Flow and Connection Details

3.1 System Endpoints

The APIs have been exposed through AWS API Gateway. The endpoint to use is of the form: <endpoint>/<service>/<operation>.

Where,
<endpoint> will be provided in a Postman Collections file,
<service> and <operation> are defined in subsequent pages.

Also note that the Stub Requests for Quickstart are provided in a separate Postman Collections (ref: https://www.getpostman.com/collection) file. You are highly recommended to install the chrome app in order to use this starter kit (download here: https://www.getpostman.com/).

3.2 Accessing APIs

APIs are accessed by triggering a HTTP POST call to the respective endpoints in 3.1, along with the correct Service and Operation values in Section 4.

All request/response are in JSON format.

The Postman Collection that will be provided to complement this document contains sample requests for the entire suite of APIs. Modify the values where applicable, and you are ready to test the services.

3.3 Overall Flow

The APIs that have been exposed can be split into 3 categories: Krisflyer, Booking, and Checkin.

Krisflyer service allows you to retrieve the profile information for dummy Accounts created in our Krisflyer programme (in test environment). The more interesting and challenging ones are the Booking and Checkin flow. Here the session, and the order of API calls play an important role. Given below are 2 possible orders of calls that can be made:

Flow 1	Flow 2
---------Booking---------- **Flight Search** **Get Fare** **Extend Session** **Create PNR** **GetPNR** (for verification) **Confirm PNR** ------ session ends here -------- **GetSeatMap** **SelectSeat** ---------Checkin---------- **GetPassenger** (retrieve new IDs) **UpdatePassenger** (must update passport/birthday details) **Checkin** **CancelCheckin** (to reuse same passenger for checkin) **GetBoardingpass** (after update and checkin)	---------Booking---------- **Flight Search** **Get Fare** **Extend Session** **Create PNR** **GetPNR** (for verification) **GetSeatMap** **SelectSeat** **Confirm PNR** **GetPNR** (for verification) ------ session ends here -------- **GetPassenger** (retrieve new IDs) **UpdatePassenger** (must update passport/birthday details) **Checkin** **CancelCheckin** (to reuse same passenger for checkin) **GetBoardingpass** (after update and checkin)

Flow Considerations:

1. A sessionId is returned in the response from `FlightSearch` which will be used for all subsequent calls that are part of the session. **All calls that are part of the session are only to be called once.** Any more than one call per operation, and you would have to start over. These are the calls that are "outside" the session:
    a. Get PNR
    b. Get Seat Map
    c. Select Seat
2. Since the above calls are outside the session, it is possible that in flow 2 we exceed the time limit of the session (~15mins) and hence we have to make periodic calls to `ExtendSession`
3. A temporary booking number (PNR) is created after the `CreatePNR` call, but this is transient until the `ConfirmPNR` call is made. If session times out before `ConfirmPNR`, the PNR is lost, and we have to restart from `FlightSearch`.
4. Once ConfirmPNR is successfully executed, the PNR becomes permanent and the session is no longer a concern.
5. Only after a confirmed booking can the checkin services be used.
6. The `flightID`s and `passengerID`s will be different in the Checkin Service calls. Please use the `GetPassenger` operation to get the IDs to use for the checkin services.

Data Considerations:

1. All Codes (Country, City, Airport, etc) follow the IATA standard.
2. Test flights with dummy data have been setup for this AppChallenge. They are:
    a. SQ366 – SIN->FCO on 1st, 7th, 15th, 22nd and 29th Nov.

b. SQ365 – FCO->SIN on 1st, 7th, 15th, 22nd and 29th Nov.

Note that only the above flight can be searched/booked/checked-in for the AppChallenge. Any deviation would result in a blanket "Invalid Request Body" response.

3. For the Krisflyer Service, 10 dummy accounts have been created for this AppChallenge. They are:
    a. 8987011927 (tier: KFLY)
    b. 8987011938 (tier: KFLY)
    c. 8987011949 (tier: KFES)
    d. 8987011916 (tier: KFEG)
    e. 8987011953 (tier: QPP)
    f. 8987011964 (tier: QPP)
    g. 8987011975 (tier: TPP)
    h. 8987011986 (tier: LPP)
    i. 8987011997 (tier: KFTP)
    j. 8987011905 (tier: ESLP)

    Any other KFNumber requested will also result in a blanket "Invalid Request Body" error.

4. To fully understand the Krisflyer Profile response, research further and fully understand the Krisflyer programme. This link should help you get sufficient information: https://www.singaporeair.com/en_UK/sg/ppsclub-krisflyer/
5. Since the document is long, suggestion is to read the "Description" of each operation and then dive deeper into the ones that you are interested.

Errors Considerations:

1. An error message suggesting that the session has timed out necessitates that you restart from flight search
2. Almost all of the request validation will result in a blanket "Invalid Request Body" error message. Hence you are strongly recommended to use the Postman collection given to you by modifying appropriate values before beginning to convert it into code.

4 API List & Specifications

Booking APIs

API	Description
flight/search	An API-version of the usual Flight Search that happens in the Website. Returns possible routes from origin to destination and also the total fares. For return-journeys, all accepted combinations of onward and return flights is also returned (allowed matching is fixed).
flight/getfare	Used to retrieve the breakdown of fares for the chosen flight. This call is important for the back-end systems to register the fare for the selected flight combination.
pnr/create	Creates a temporary PNR to store the passenger information. This PNR is transient and will be invalidated within roughly 10 mins, unless the PNR is confirmed, or the session is extended (refer to calls below).
pnr/get	Retrieves the information in the PNR, such as passenger information, flight information, and other services bought by the passenger.
seat/getmap	This call retrieves the seat configuration of the aircraft. It includes information such as the positions of the seat (row/column), characteristics (window, near-exit, etc), facilities (galley, lavatories, staircases, exits)
seat/select	This call reserves a seat subject to availability (indicated in the seatmap above).
pnr/confirm	Completes the booking, and the PNR becomes permanent. The PNR can then be used later to check-in, update seat choice or simply retrieve the booking information. Once this call is successful, session is no longer a concern.
misc/extendsession	Since the transaction is time-bound, use this call to extend the timeout. Failure to make this call in a timely fashion (every 10 mins) will result in loss of the transaction and the need to start-over (from flight search).

Checkin APIs

API	Description
checkin/getpassenger	Retrieves the passenger information (taken from PNR)
checkin/updatepassenger	Allows updating of passport details. This call is mandatory for subsequent calls to work reliably.
checkin/checkin	Checks-in the passenger(s)
checkin/cancelcheckin	Cancels the checkin, allowing for check-in again
checkin/getboardingpass	Provides the necessary information to create a boardingpass, including the qrcode/barcode. Note: Call will succeed only if passenger is checked-in and passport is updated

Krisflyer APIs

API	Description
krisflyer/getprofile	Retrieves the member's details such as name, contact details as well as a summary of their loyalty miles (kfMiles/ppsMiles).

4.1 Flight Search

Service Name	flight		
API Name	search		
Description	This API is used to search for flights based on airport and departure date. Only Singapore (SIN) to Rome (FCO) and Rome (FCO) to Singapore (SIN), on dates: 1st, 8th, 15th, 22nd and 29th November are available. Any other combination will result in "invalid request" error		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
itineraryDetails	JSON Array	No	1 or 2 elements depending on whether it is one-way or two-way
originAirportCode	String	No	"SIN" or "FCO"
destinationAirportCode	String	No	"SIN" or "FCO"
departureDate	String	No	Only these dates will return values: "2017-11-01", "2017-11-08", "2017-11-15", "2017-11-22", "2017-11-29"
cabinClass	String	No	"F"/"J"/"S"/"Y"
adultCount	String	No	Any integer below 9
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
sessionId	String	Used to keep session. Use this value to fill subsequent calls during the booking flow	
currency	JSON Object	Provides details on the base currency used for this booking	
code	String	3-letter currency-code (eg: SGD)	
name	String	Currency name in full (eg: Singapore Dollar)	
precision	Integer	Number of decimal places when the currency is used	
tripType	String	"O" – One-way ; "R" – Return	
flights	JSON Array	A List of possible flights to choose from	
originAirportCode	String		
destinationAirportCode	String		
departureDateTime	String		
arrivalDateTime	String		
legs	JSON Array	More information on each leg of flight. Useful when a flight stops over at a location	
flightNumber	String		
originAirportCode	String	3-letter code	
destinationAirportCode	String	3-letter code	
departureTerminal	String		
arrivalTerminal	String		
departureDateTime	String	"yyyy-MM-dd HH:mm:ss"	
arrivalDateTime	String	"yyyy-MM-dd HH:mm:ss"	
operatingAirline	JSON Object	The actual airline that operates the flight in this leg	
code	String	2-letter airline code (eg: SQ)	
name	String	Airline name in full (eg: Singapore Airlines)	
marketingAirline	JSON Object	The airline through which the ticket is bought. Read more on code-share flights to understand marketing vs operating flight	

code	String	2-letter airline code (eg: SQ)
name	String	Airline name in full (eg: Singapore Airlines)
aircraft	JSON Object	Information on the vehicle
code	String	3-digit aircraft code
name	String	Name of the Aircraft Model
flightDuration	Integer	in seconds
layoverDuration	Integer	In seconds
tripDuration	Integer	Total time for the journey in seconds
defaultRecommendationID	Integer	Lowest Fare Recommendation
Recommendations	JSON Array	Since **each** flight can have multiple RBDs (eg: Economy Flexi, Economy Flexi Saver), these contain information on each one. For return flights, it contains matching variations of both flights.
recommendationID	Integer	Unique identifier
segmentBounds	JSON Array	Number of elements in array will match number of elements in the “itineraryDetails” in request
segments	JSON Array	Structure mirrors that of “flights” defined earlier in response
segmentID	Integer	Matches flightID in “flights” defined earlier in response
displayLastSeat	Boolean	“True” if very few seats are available
numOfLastSeats	Integer	Threshold used to set the “displayLastSeat” flag. If available seats are fewer than this number, flag above is set.
legs	JSON Array	Matches with “legs” inside the “flights” field
sellingClass	String	RBD value
cabinClass	String	“F”/”J”/”S”/”Y”
fareClass	String	Can be ignored
fareFamily	String	Unique String identifying the fare-family. More details on Farefamily can be found under “fareFamilies” later in the response
fareSummary	JSON Object	Details on the fare
fareTotal	JSON Object	Combined fare details for all passengers
totalAmount	String	
AmountWihoutTax	String	
Tax	String	
fareDetailsPerAdult	String	Adult-wise split of the above “fareTotal” details
totalAmount	String	
AmountWihoutTax	String	
Tax	String	
fareFamilies	String	More details on Farefamilies.
fareFamily	String	Value used to match with the rest of the response
fareFamilyCode	String	
fareFamilyName	String	The value that is displayed on screen to the passenger
cabinClass	String	“F”/”J”/”S”/”Y” that this fare belongs to
cabinClassName	String	Eg: “Economy”
airports	JSON Array	More details on the airports
airportCode	String	3-letter code
airportName	String	
cityCode	String	2-letter code
cityName	String	
countryCode	String	2-letter code
countryName	String	

4.2 GetFare

Service Name	flight		
API Name	getfare		
Description	Retrieves details breakdown of fare. Also used internally to capture the fare to use while billing. The information for this call is taken from the "Flight Search" (section 4.1) response. Retrieve a detail fare breakdown based on preferences and flights.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
sessionId	String	No	"SessionId" copied over from the response of Flight Search call earlier
request	JSON Object	No	
recommendationDetails	JSON Array	No	Information from the response in "Flight Search" Call
recommendationId	Integer	No	Id of the specific recommendation we are interested to get fares for
flightIds	Integer Array	No	SegmentIds from each "SegmentBound" object under the specific recommendation with ID given above. Note: it is usually [0] for oneway and [0,0] for return.
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
message	String	Used only if status is "FAILED". Ignored otherwise.	
sessionId	String	SessionId to use for subsequent calls. Note that it is the same as in request.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
fareDetails	JSON Object		
currency	JSON Object		
code	String		
name	String		
precision	Integer	Number of decimal places used in the currency values	
adultCount	String		
fareAmount	String		
taxAmount	String		
totalAmount	String		
pricePerBound	JSON Array	Flight wise split-up of the fare.	
fareAmount	String		
taxAmount	String		
totalAmount	String		
fareFamilDetails	JSON Object		
code	String		
description	String	RBD in full (eg: Economy Flexi Saver)	
travellerListPrice	JSON Object		
travellerType	String	"ADT"	
fareAmount	String		
taxAmount	String		
totalAmount	String		
taxList	JSON Array	List of all the taxes and corresponding fare breakdown	
code	String	type of tax code	

description	String	Description of tax
category	String	Ignore
amount	String	amount of money paid for the tax component
pricePerBound	JSON Array	Price split-up by flight. (1 for one-way and 2 for return flight)
fareAmount		
taxAmount		
totalAmount		

4.3 Create PNR

Service Name	pnr		
API Name	create		
Description	Creating a transaction. If PNR is not created within session time, the PNR will be removed.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
tripType	String	No	"O"/"R"
travellerDetails	JSON Array	No	Number of elements must match number of passengers
title	String	No	
firstName	String		
emailAddress	String		
isLeadPax	String		
contactDetails	String		
type	String	No	"MOBILE"/"HOME"/"OFFICE"
countryCode	String	No	2-digit code
number	String	No	phone number (random)
sessionId			
Output Parameters	**Data Type**	**Description**	
Status	String	"SUCCESS" / "FAILED"	
Code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
sessionId	String	Used to keep session. Use this value to fill subsequent calls during the booking flow	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
pnrDetails	JSON Object		
pnr	String	6-alphanumeric	
currency	String	3-letter code	
travellerDetails	JSON Array		
passengerID	String	Used in the "confirm" pnr	
type	String	"ADT" (Adult)	
title			
firstName	String		
lastname	String	"Appchallenge"	
emailAddress	String		
passengerIDForSeat	String	ID to use when calling Select Seat service	
leadPassenger	Boolean	Indicates the lead passenger in PNR	
segmentIdList	String Array	Numbers to identify the various flights being taken. The order matches the order of the booking (eg: 1 -> onward flight, 2-> return flight).	

4.4 Get PNR

Service Name	Pnr		
API Name	Get		
Description	Retrieve passenger records and flights information based on PNR Only Singapore (SIN) to Rome (FCO) and Rome (FCO) to Singapore (SIN), on dates: 1st, 8th, 15th, 22nd and 29th November are available. Any other combination will result in "invalid request" error		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
pnr	String	No	PNR returned in the Create PNR call
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
pnr	String		
currency	String	3-letter code	
passengers	JSON Array		
passengerIDForSeat	String	ID to use for Seat services	
passengerType	String	"ADT"	
lastName	String		
firstName	String		
title	String		
ticketNumber	String	populated if "pnr/get" is called after booking	
gender	String	"M" / "F" if it was provided at "pnr/create"	
dateOfBirth	String		
additionalPassengerDetails	JSON Object	Contains regulatory documents such as passport details. It can be added using "checkin/updatepassenger" call	
documentType	String	eg: "PASSPORT"	
documentNumber	String		
expiryDate	String		
issueDate	String		
issuingCountry	String		
seatDetails	JSON Array	One element per flight	
flightID	String		
seatNumber	String	Row and Column (eg: "50A")	
flights	JSON Array	Flight information	
flightSegment	JSON Array	Flight Segments (useful when there are stopovers)	
marketingAirlineCode	String	2-letter code The airline through which the ticket is bought. Read more on code-share flights to understand marketing vs operating flight	
marketingAirlineName	String		
origin	JSON Object	Origin details for this segment of flight	
airportCode	String	3-letter code	
airportName	String		
terminalNumber	String		
countryCode	String	2-letter code	
destination		Destination details for this segment of flight	

airportCode	String	3-letter code
airportName	String	
terminalNumber	String	
countryCode	String	2-letter code
scheduledDepartureTime	String	“yyyy-MM-dd HH:mm:ss.SSS”
scheduledArrivalTime	String	“yyyy-MM-dd HH:mm:ss.SSS”
estimatedDepartureTime	String	“yyyy-MM-dd HH:mm:ss.SSS”. Differs from scheduled if delay is expected
estimatedArrivalTime	String	“yyyy-MM-dd HH:mm:ss.SSS”.
cabinClass	String	Full name (eg: “ECONOMY”)
sellingClass	String	RBD
origin	JSON Object	Origin details for the entire flight
airportCode	String	3-letter code
airportName	String	
terminalNumber	String	
countryCode	String	2-letter code
destination		Destination details for this segment of flight
airportCode	String	3-letter code
airportName	String	
terminalNumber	String	
countryCode	String	2-letter code
flightNumber	String	“366”/”365”
flightId	String	Unique ID for the flight. Useful for seat map retrieval or seat selection

4.4 Get Seat Map

Service Name	seat		
API Name	getmap		
Description	Retrieves the seat map details for the flight specified in the request. The best way to understand this response is to compare it to the seat map picture you see during a real booking with SQ. You may also refer to the following document, although the facilities may not be exactly the same: http://www.singaporeair.com/saar5/pdf/OurFleet/airbus380-4class.pdf Note: some seats are blocked due to company policy and are allocated at random at checkin counter.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
recordLocator	String	No	6 alpha-numeric string (PNR). Uniquely identifies a booking
flightID	String	No	Unique identifier for the flight
flightNumber	String	No	Strictly only “366” or “365”
origin	String	No	3-letter IATA code – SIN, LHR
destination	String	No	3-letter IATA code – HKG, SFO
departureDate	Date	No	W3C date time format: YYYY-MM-DDThh:mm Setting hh:mm to 00:00 will work for all flights.
cabinClass	String	No	“F”/”J”/”S”/”Y”. Must match the cabin class used while booking to get relevant results.
pointOfSales	String	Yes	“FCO”, or “SIN” based on the first boarding point (chronologically the earliest airport where passenger boards the flight)
Output Parameters	**Data Type**	**Description**	
status	String	“SUCCESS” / “FAILED”	
code	Number	200 if status is “SUCCESS”. Refer to Section 5 for possible error codes	
message	String	Used only if status is “FAILED”. Ignored otherwise.	
response	**JSON Object**	empty if status is “FAILED”; displayed otherwise	
pnr	String		
passengers	**JSON Array**		
passengerID	String	Unique identifier for a passenger	
lastName	String		
firstName	String		
title	String		
seatSelected	JSON Object		
seatNumber	String	Seat number, e.g. “31A”	
flight	**JSON Object**		
flightID	String	Unique identifier for a flight	
flightNumber	String	“365” or “366”	
operatingAirlineCode	String	2-letter code	
operatingAirlineName	String		
marketingAirlineCode	String	2-letter code	
marketingAirlineName	String		
cabinClass	String	“F”/”J”/”S”/”Y”	
origin	**JSON Object**		
airportCode	String	3-letter code	

airportName	String	
airportTerminal	String	
cityName	String	Full name
destination	**JSON Object**	
airportCode	String	3-letter IATA code – SIN, LHR
airportName	String	London Heathrow
airportTerminal	String	
cityName	String	London
equipment	**JSON Object**	Details about the aircraft; the actual vehicle
acv	String	3-digit code to identify the model
aircraft	String	Full name of aircraft
cabinCapacity	String	Alphanumeric string indicating cabin capacities in the aircraft. Eg: "Y333S36J60F12" -> 333 in Economy, 60 in Business, etc
wingStartRow	String	Row at which the wing starts. Only returned if it is within the cabin class that was requested.
wingEndRow	String	Row at which the wing ends. Only returned if it is within the cabin class that was requested.
cabinAmenities	**JSON Array**	
power	String	“110V AC”
screen	String	“10.6-inch LCD screen”
bed	String	“Full-flat bed”
seat	String	“19-inch width”
compartment	**JSON Array**	Separation of seats within a given cabin class
designator	String	Unique identifier (Eg: “C”)
location	String	“M” – main deck; “U” – upper deck
startRow	String	First row number of compartment
endRow	String	Last row number of compartment
column	**JSON Array**	Generic column details for all seats in the compartment
designator	String	Column value (“A” – “K”)
characteristics	String	“WAM”. W – Window; A – Aisle; M – Middle. If it is a single seat (usually in business or first class), it can be “WA”
row	**JSON Array**	All seat information. Includes facilities defined in a similar manner to the seats.
number	String	row number (eg: “41”). Note that this number can repeat for multiple rows. If it does, all but one are purely facility definitions, and only one contains the actual seats. Useful when showing compartments separately to know where a facility at the boundaries belongs.
seat	**JSON Array**	
column	String	“A” – “K”
occupied	Boolean	Indicates if the seat is available for selection.
characteristics	String	“WAMBCEHO”. W – Window; A – Aisle; M – Middle; B – Bassinet; C – Chargeable; E – Emergency Exit; H – Handicap; K – Windowless; O – Bubble
facility	**JSON Array**	Defined using either column or location
column	String	“A” – “K”
location	String	“LCR” L – Left; R – Right; C – Centre
type	String	“LGSDE”

		L – Lavatory; G – Galley; S – Stairs; D – Accessible Lavatory; E – Exit-Door;

4.5 Select Seat

Service Name	seat		
API Name	select		
Description	Used to select seats for all passengers in a specific flight.		
Input Parameters	Data Type	Optional	Description
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
pnr	String	No	
passengerSeatInfo	JSON Array	No	Specifies which seat is selected for which passenger
passengerID	String	No	Use the "passengerIDForSeat" from the response of "pnr/create" or "pnr/get"
seatNumber	String	No	Row and Column (eg: 50A)
flightID	String	No	
destination	String	No	"SIN" or "FCO"
origin	String	No	"SIN" or "FCO"
Output Parameters	Data Type	Description	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
segment	JSON Object		
passengerSeatInfo	JSON Object		
passengerID	String		
seatNumber	String		
seatStatus	String	"Confirmed"	
flightID	String		

4.6 Extend Session

Service Name	misc		
API Name	extendsession		
Description	Used to extend the session timeout by roughly 15 mins. Do note that if the session times out, you have start over from flight search.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
sessionId	String	No	session to extend the timeout for
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
message	String	Used only if status is "FAILED". Ignored otherwise.	
sessionid	String	replicated from request	

4.6 Confirm PNR

Service Name	pnr		
API Name	confirm		
Description	Confirms the booking and ends the need for session. Most of the details in the request are replicated from the "pnr/get" call. After this call the PNR becomes permanent and can be retrieved using "pnr/get" or updated using "seat/select" anytime. Before this call, if the 15mins of session times out, the booking is lost. Note: Generate a random card **MasterCard** number here to use for this request. To change the card type, click on 'American Express': https://saijogeorge.com/dummy-credit-card-generator/		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
pnr	String	No	
travellerDetails	JSON Object	No	
isLeadPassenger	Boolean	No	
contactDetails	JSON Object	No	
type	String	No	"MOBILE"/"HOME"/"OFFICE"
value	String	No	
title	String	No	
travellerID	String	No	
firstName	String	No	
addressDetails	JSON Object	No	
address	String	No	
city	String	No	Full name
state	String	No	
zipCode	String	No	
countryCode	String	No	2-letter code
paymentDetails	JSON Object	No	
payeeName	String	No	
accountNumber	String	No	Generate from weblink in Description above. Generate a **MasterCard** number.
expiryDate	String	No	"yyMM"
cvvNumber	String	No	any 3 digits
amount	String	No	use the total fare from the "flight/getfare" call
currencyCode	String	No	Use the currency returned from "pnr/get"
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
pnr	String		
ticketList	JSON Array	Mapping between passengerID and ticketNumber. In most cases the ticketNumber is the same for all passengers	
ticketNumber	String		
passengerID	String		
receiptDetails	JSON Array	Insurance payment receipts. Note that for some locations insurance is added in by default.	
receiptNumber	String		

currencyCode	String	
price	String	
insuranceDetails	JSON Array	
confirmationNumber	String	
issuanceSuccess	Boolean	

4.7 Krisflyer Profile

Service Name	krisflyer		
API Name	getprofile		
Description	Returns the profiles of the Kriflyer members. 10 KF accounts have been created for the AppChallenge. They are given in the Description column below.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
krisflyerNumber	String	No	10 numeric string. Uniquely identifies a member. Please use only one of the following numbers: 8987011927 (KFLY) 8987011938 (KFLY) 8987011949 (KFES) 8987011916 (KFEG) 8987011953 (QPP) 8987011964 (QPP) 8987011975 (TPP) 8987011986 (LPP) 8987011997 (KFTP) 8987011905 (ESLP)
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
Message	String	Used only if status is "FAILED". Ignored otherwise.	
response	**JSON Object**	empty if status is "FAILED"; displayed otherwise	
lastName	String		
firstName	String		
title	String		
gender	String	"M", "F"	
dateOfBirth	Date	W3C date time format: YYYY-MM-DD	
loyaltyTierCode	String	1-letter code	
loyaltyTierName	String		
passport	JSON Object		
passportNumber	String		
expiryDate	Date	W3C date time format: YYYY-MM-DD	
issuingCountry	String	3-letter country code	
nationality	String	3-letter country code	
address	JSON Array		
type	String	"Home" / "Business" / "Unspecified"	
line1	String		
line2	String		
line3	String		
line4	String		
city	String		
state	String		
country	String	3-letter country code	
postalCode	String		
contact	JSON Array		
type	String	"Home" / "Mobile" / "Business" / "Unspecified"	
countryCode	String	Two digit country code	

areaCode	String	
phoneNumber	String	
accountSummary	JSON Object	
memberSince	Date	W3C date time format: YYYY-MM-DD
cardExpiryDate	Date	Refers to the expiry date of the physical card. A replacement has to be made after this date. W3C date time format: YYYY-MM-DD
kfMiles	String	Currently available KfMiles
eliteMiles	String	Currently available elite miles
eliteRequalifiedBy	String	Deadline to retain current tier
eliteRequalifiedMilesRequired	String	Miles to accrue by above deadline to retain current tier.
ppsMiles	String	
ppsValues	String	
ppsReserveValues	String	
ppsRequalifiedBy	String	
ppsRequalifiedMilesRequired	String	

4.8 Checkin GetPassenger

Service Name	checkin		
API Name	getpassenger		
Description	Retrieves information about the passenger and the journey he/she is going to undertake. The information from this call will be used for subsequent checkin calls.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
krisflyerNumber	String	No	10 numeric string. Uniquely identifies a member
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS". Refer to Section 5 for possible error codes	
message	String	Populated and displayed only if status is "FAILED"	
response	JSON Object		
partiallyCheckedIn	Boolean		
recordLocator	String	Refers to PNR The 6-char string identifying the PNR.	
flights	JSON Array		
flightIDs	String Array	Refers to DIDs. These are the IDs to use for all checkin related calls.	
origin	JSON Object		
airportCode	String	3-letter IATA code – SIN, LHR The 3 digit code determining the airport	
airportName	String	The 3 digit code determining the airport , London Heathrow	
airportTerminal	String	The terminal	
cityName	String	The city name	
destination	JSON Object		
airportCode	String	3-letter IATA code – SIN, FCO The 3 digit code determining the airport	
airportName	String	The 3 digit code determining the airport , London Heathrow	
airportTerminal	String	The terminal	
cityName	String	The city name	
operatingAirline	JSON Object		
airlineCode	String	The two-char carrier code	
airlineName	String	The name of the carrier	
flightNumber	String	3 or 4 digits identifying the flight number	
marketingAirline	JSON Object		
airlineCode	String	The two-char carrier code	
airlineName	String	The name of the carrier	
flightNumber	String	3 or 4 digits identifying the flight number	
scheduledDepartureDateTime	String	W3C date time format: YYYY-MM-DDThh:mm Scheduled departure datetime of the flight.	
scheduledArrivalDateTime	String	W3C date time format: YYYY-MM-DDThh:mm Scheduled arrival datetime of the flight.	
estimatedDepartureDateTime	String	W3C date time format: YYYY-MM-DDThh:mm Estimated departure datetime of the flight.	

estimatedArrivalDateTime	String	W3C date time format: YYYY-MM-DDThh:mm Estimated arrival datetime of the flight
tci	boolean	True for through check-in flights
rci	boolean	True for return check-in flights
passengers	JSON Array	
passengerID	String	Refers to UCI The passenger ID. This is the passengerID to use for all checkin calls
flightIDs	String Array	The list of flight IDs associated with the passenger
lastName	String	Last name / surname of the passenger.
firstName	String	First name / given name of the passenger.
title	String	Addressing title of the passenger.
dateOfBirth	String	W3C date format: YYYY-MM-DD Passenger's date of birth
gender	String	Gender of the passenger, given by a single char (M / F)
passengerType	String	Type of passenger (adult)
addressInfo	JSON Object	
address	String	Address of the passenger
stateName	String	Country state name
countryName	String	Country name
cityName	String	City Name
zipCode	String	Zip Code
countryOfResidence	String	Passenger's country code of residence
prCard	JSON Object	Only for USA (SSR DOCO)
number	String	Reference number of the Military card. ,
expiryDate	String	W3C date format: YYYY-MM-DD Expiry date of the PR card
issuingCountryCode	String	3-character country code, e.g. SGP
registrationCountryCode	String	3-character country code, e.g. SGP
type	String	Type of PR card. = ['UNKNOWN', 'military', 'alien', 'pr', 'reentry']
firstName	String	First name on military card.
lastName	String	Last name on military card.
passport	JSON Object	
firstName	String	First name of the passenger
lastName	String	Last name / given name of the passenger.
expiryDate	String	W3C date format: YYYY-MM-DD Passport expiry date
issuingCountryCode	String	3-character country code, e.g. SGP
nationalityCode	String	3-character country code, e.g. SGP
passportNumber	String	Passport number of the passenger.
contactDetails		
mobileNumber	String	Mobile phone number of passenger
mobileCountryCode	String	Mobile phone number country code of passenger
email	String	Email address of the passenger
nextOfKinDetails		
name	String	Next-of-kin name
relation	String	Next-of-kin relation
mobileNumber	String	Next-of-kin mobile number
mobileCountryCode	String	Next-of-kin mobile phone country code
services	JSON Array	

flightID	String	Identifier of the customer-flight pair, linking the service
passengerID	String	Refers Identifier of the customer, linking the service
cabinClass	String	The cabin class of the flight
seatSelected	String	Assigned seat on the plane. If two-digit, seat starts with a 0.
ticketNumber	String	Either 'ticketNumber' or 'recordLocator' to be mandatory The ticket number identifying the PNR
upgraded	Boolean	True if FQTU is present Whether seat is upgraded
dcsStatus	JSON Object	
boardingPassIssued	Boolean	True if boarding pass is issued
checkedIn	Boolean	True if flight is checked in
docChecksPassed	Boolean	True if ADC check is passed
regulatoryChecksPassed	Boolean	True if APP check is passed, and not overridden
spBpEnabled	Boolean	True if SP Boarding Pass indicator is enabled
ccvDocCheckRequired	Boolean	True if Credit Card needs to be verfieid
visaDocCheckRequired	Boolean	True if visa documents have to be verified
dcsIceIsBlocked	Boolean	True if DCS ICE is blocked from checking in
fqtvDetails	JSON Object	
loyaltyTierCode	Boolean	Tier code of the frequent flyer programme
loyaltyTierName	Boolean	Tier name of the frequent flyer programme
ffpAirline	String	Corresponding field in SSR FQTV Carrier providing the frequent flyer programme
ffpNumber	String	Frequent flyer number
fqtsDetails	JSON Object	
loyaltyTierCode	Boolean	Tier code of the frequent flyer programme
loyaltyTierName	Boolean	Tier name of the frequent flyer programme
ffpAirline	String	Corresponding field in SSR FQTV Carrier providing the frequent flyer programme
ffpNumber	String	Frequent flyer number

4.9 Checkin UpdatePassenger

Service Name	checkin		
API Name	updatepassenger		
Description	Used to updates the passport of the passenger. Note that after this call, both "pnr/get" and "checkin/getpassenger" will be updated with this value.		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
pnr	String	No	10 numeric string. Uniquely identifies a member
passengers	JSON Array	No	This information is replicated from the getpassenger call
passengerID	String	No	Refers to UCI The passengerID from getpassenger call
firstName	String	No	First name / given name of the passenger.
title	String	No	Addressing title of the passenger.
gender	String	No	Gender of the passenger, given by a single char (M / F)
regulatoryDocs	JSON Object	No	
expiryDate	String	No	W3C date format: YYYY-MM-DD
firstName	String	No	First name on military card.
issuingCountryCode	String	No	3-character country code, e.g. SGP
nationalityCode	String	No	3-character country code, e.g. SGP
documentNumber	String	No	Reference number of the Military card
Output Parameters	**Data Type**	**Description**	
status	String	"SUCCESS" / "FAILED"	
code	Number	200 if status is "SUCCESS".	
message	String	Used only if status is "FAILED". Ignored otherwise.	

4.10 Checkin Checkin

Service Name	checkin		
API Name	checkin		
Description	Checks in the given passengers to the given flights.		
Input Parameters	Data Type	Optional	Description
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
flights	JSON Array	No	replicated from getpassenger call
flightIDs	String Array	No	Refers to DIDs
originAirport	JSON Object	No	
airportCode	String	No	3-letter IATA code – SIN, LHR
airportTerminal	String	No	The terminal
destinationAirport	JSON Object	No	
airportCode	String	No	3-letter IATA code – SIN, LHR
airportTerminal	String	No	The terminal
flightNumber	String	No	“366” or “365”
departureDateTime	String	No	W3C date time format: YYYY-MM-DDThh:mm
passengers	JSON Array	No	replicated from getpassenger call
passengerID	String	No	Refers to UCI
firstName	String	No	First name of the passenger
title	String	No	Addressing title of the passenger.
gender	String	No	Gender of the passenger, given by a single char (M / F)
flightIDs	String Array	No	The list of flight IDs associated with the passenger
Output Parameters	Data Type	Description	
status	String	“SUCCESS” / “FAILED”	
code	Number	200 if status is “SUCCESS”.	
Message	String	Used only if status is “FAILED”. Ignored otherwise.	

4.11 Checkin CancelCheckin

Service Name	checkin		
API Name	cancelcheckin		
Description	Cancels the check-in for the given passengers in the given flights.		
Input Parameters	Data Type	Optional	Description
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
flights	JSON Array	No	replicated from getpassenger call
flightIDs	String Array	No	Refers to DIDs
originAirport	JSON Object	No	
airportCode	String	No	3-letter IATA code – SIN, LHR
airportTerminal	String	No	The terminal
destinationAirport	JSON Object	No	
airportCode	String	No	3-letter IATA code – SIN, LHR
airportTerminal	String	No	The terminal
flightNumber	String	No	“366” or “365”
departureDateTime	String	No	W3C date time format: YYYY-MM-DDThh:mm
passengers	JSON Array	No	replicated from getpassenger call
passengerID	String	No	Refers to UCI
firstName	String	No	First name of the passenger
title	String	No	Addressing title of the passenger.
gender	String	No	Gender of the passenger, given by a single char (M / F)
flightIDs	String Array	No	The list of flight IDs associated with the passenger
Output Parameters	Data Type	Description	
status	String	“SUCCESS” / “FAILED”	
code	Number	200 if status is “SUCCESS”.	
Message	String	Used only if status is “FAILED”. Ignored otherwise.	

4.12 Checkin GetBoardingPass

Service Name	checkin		
API Name	getboardingpass		
Description	Retrieves all the necessary information for printing a boardingpass. Note that after this call, the passenger's checkin status cannot be cancelled		
Input Parameters	**Data Type**	**Optional**	**Description**
clientUUID	String	No	Unique string representing this call
request	JSON Object	No	
pnr	String	No	
passengers	JSON Array	No	replicated from getpassenger call
passengerID	String	No	Refers to UCI
flightIDs	String Array	No	The list of flight IDs associated with the passenger

Output Parameters	**Data Type**	**Description**
status	String	"SUCCESS" / "FAILED"
code	Number	200 if status is "SUCCESS". Refer to Section 5 for possible error codes
message	String	Populated and displayed only if status is "FAILED"
response		
recordLocator	String	Either 'ticketNumber' or 'recordLocator' to be mandatory The 6-char record locator identifying the PNR
passengers	JSON Array	
passengerID	String	Unique identifier for a passenger The passenger ID
lastName	String	Last name / surname of the passenger.
firstName	String	First name of the passenger
title	String	Addressing title of the passenger.
adult	Boolean	True if adult
infant	Boolean	True if infant
flightIDs	String Array	The list of flight IDs associated with the passenger
flights	JSON Array	
flightID	String	Unique identifier for a flight
flightNumber	String	3 or 4 digits identifying the flight number
operatingAirlineCode	Boolean	2-letter IATA code – SQ, MI, TZ, TR
operatingAirlineName	String	
marketingAirlineCode	String	2-letter IATA code – LH, VA, VX (Populated if available)
marketingAirlineName	String	(Populated if available)
origin	JSON Object	
airportCode	String	3-letter IATA code – SIN, LHR
airportName	String	Singapore Changi
airportTerminal	String	The terminal
cityName	String	The city name
destination	JSON Object	
airportCode	String	3-letter IATA code – SIN, LHR
airportName	String	London Heathrow
airportTerminal	String	The terminal
cityName	String	The city name
	String	W3C date time format: YYYY-MM-DDThh:mm
	String	W3C date time format: YYYY-MM-DDThh:mm
estimatedDepartureDateTime	String	W3C date time format: YYYY-MM-DDThh:mm (Populated if available)

estimatedArrivalDateTime	String	W3C date time format: YYYY-MM-DDThh:mm (Populated if available)
tripDuration	String	W3C date time format: hh:mm, e.g. 3:45, 17:00
cabinClass	String	The cabin class of the flight
sellingClass	String	1 alpha
aircraftType	String	e.g. “Boeing 777-300ER”
services	JSON Array	
passengerID	String	Unique identifier for a passenger The passenger ID
flightID	String	Unique identifier for a flight
ticketNumber	String	Either ‘ticketNumber’ or ‘recordLocator’ to be mandatory The ticket number identifying the PNR
seat	JSON Object	
seatSelected	String	Seat number – 04A, 44G
deckInfo	String	Applicable for double deck config – Main, Upper
boardingPass	JSON Object	
mFormatString	String	IATA Resolution 792 v5 compliant
base64Image	String	base64 representation of image – QR for Mobile, PDF417 for Web
nameOnBoardingPass	String	
loungeText	String	(Populated if available)
boardingDateTime	String	W3C date time format: YYYY-MM-DDThh:mm
boardingGate	String	(Available on -3 Hrs STD, thereafter)
boardingGroup	String	For segregation of boarding groups of passengers
regulatoryMessage	String	
airlineUse	String	
TSAPrecheck	Boolean	True if BP fields are signed.
ffpDetails	JSON Object	
krisFlyer	Boolean	True if ffpAirlineAccrual or ffpAirlineServicing is SQ
alliance	Boolean	True if allianceTierCode is present
ffpAirlineAccrual	String	Corresponding field in SSR FQTV
ffpNumberAccrual	String	Corresponding field in SSR FQTV
loyaltyTierCodeAccrual	String	Corresponding field in SSR FQTV
loyaltyTierNameAccrual	String	Corresponding field in SSR FQTV
ffpAirlineServicing	String	Corresponding field in SSR FQTS
ffpNumberServicing	String	Corresponding field in SSR FQTS
loyaltyTierCodeServicing	String	Corresponding field in SSR FQTS
loyaltyTierNameServicing	String	Corresponding field in SSR FQTS
allianceTierCode	String	Either from validated FQTV or FQTS
allianceTierName	String	Either from validated FQTV or FQTS
partner	Boolean	True is SK PTNR is present
partnerDescription	String	Parse free text field of SK PTNR